Ingénierie système et Sûreté de fonctionnement : Méthodologie de synchronisation des modèles d'architecture et d'analyse de risques

Anthony Legendre

université PARIS-SACLAY

NNT : 2017SACLC083

CentraleSupélec

Ingénierie système et Sûreté de fonctionnement : Méthodologie de synchronisation des modèles d'architecture système et d'analyse de risques

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Ecole CentraleSupélec

École doctorale n°573 INTERFACES : approches interdisciplinaires, fondements, applications et innovation
Spécialité de doctorat : l'ingénierie des systèmes complexe

Thèse présentée et soutenue à Palaiseau, le 15 décembre 2017, par

Anthony Legendre

Composition du Jury :

Eric BONJOUR Professeur des Universités, Université de Lorraine – Laboratoire ERPI	Président du jury
Abdessamad KOBI Professeur des Universités, ISTIA – Laboratoire LARIS	Rapporteur
Eric NIEL Professeur des Universités, INSA Lyon – Département MIS	Rapporteur
Frédéric BOULANGER Professeur des Universités, CentraleSupélec – Laboratoire LRI	Examinateur
Antoine RAUZY Professeur des Universités, NTNU – Département MTP	Directeur de thèse
Agnès LANUSSE Docteur, CEA LIST – Laboratoire LISE	Encadrante CEA
Marie Véronique SERFATY Docteur, DGA – Responsable domaine scientifique I2R	Invitée

Thèse de doctorat

université PARIS-SACLAY

NNT : 2017SACLC083

Thèse de doctorat

Ingénierie système et Sûreté de fonctionnement : Méthodologie de synchronisation des modèles d'architecture système et d'analyse de risques

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Ecole CentraleSupélec

École doctorale n°573 INTERFACES : approches interdisciplinaires, fondements, applications et innovation
Spécialité de doctorat : l'ingénierie des systèmes complexe

Thèse présentée et soutenue à Palaiseau, le 15 décembre 2017, par

Anthony Legendre

Composition du Jury :

Eric BONJOUR Professeur des Universités, Université de Lorraine – Laboratoire ERPI	Président du jury
Abdessamad KOBI Professeur des Universités, ISTIA – Laboratoire LARIS	Rapporteur
Eric NIEL Professeur des Universités, INSA Lyon – Département MIS	Rapporteur
Frédéric BOULANGER Professeur des Universités, CentraleSupélec – Laboratoire LRI	Examinateur
Antoine RAUZY Professeur des Universités, NTNU – Département MTP	Directeur de thèse
Agnès LANUSSE Docteur, CEA LIST – Laboratoire LISE	Encadrante CEA
Marie Véronique SERFATY Docteur, DGA – Responsable domaine scientifique I2R	Invitée

SOMMAIRE

REMERCIEMENTS

« La thèse de doctorat représente un travail s'inscrivant dans la durée, et pour cette raison, constitue le fil conducteur d'une tranche de vie de son auteur, parfois au crépuscule de la candeur étudiante, et souvent à l'aube de la maturité scientifique. De nombreuses personnes se retrouvent ainsi de manière fortuite ou non, pour le pire ou le meilleur, entre le doctorant et son doctorat. Ce sont certaines de ces personnes que j'aimerais mettre en avant dans ces remerciements. » (Morgan David)

Je remercie chaleureusement toutes les personnes qui m'ont aidé, soutenu, conseillé, suivi pendant l'élaboration de ma thèse.

Je souhaite remercier tout particulièrement mon directeur de thèse Monsieur le professeur Antoine Rauzy, pour son fort intérêt, son amour des sciences, sa pédagogie et ses nombreux conseils durant ma thèse. Je suis ravi d'avoir travaillé en sa compagnie, malgré la distance géographique, il a toujours su se rendre disponible et être une source d'inspiration dans mes travaux.
Je remercie également chaleureusement mon encadrante CEA, le docteur Agnès Lanusse pour son dévouement à la recherche, sa passion dans son travail et sa présence quotidienne. Agnès a toujours su opérer activement pour faire en sorte que la thèse se déroule sans embûche.

Merci au personnel de la DGA pour les rencontres effectuées et les riches discussions qu'elles ont amenées. Je les remercie également pour le suivi de l'avancement de mes travaux.

Je remercie également le docteur Sébastien Gérard de m'avoir permis de faire ma thèse dans son laboratoire. Ce travail n'aurait pas été possible sans le co-financement de mon contrat par le CEA LIST et la DGA, qui m'a permis de me consacrer sereinement à l'élaboration de cette thèse.

Je remercie l'ensemble des membres de mon jury : Monsieur le professeur Abdessamad Kobi, Monsieur le professeur Eric Niel, Monsieur le professeur Eric Bonjour et Monsieur le professeur Frédéric Boulanger.

Je remercie le personnel de l'école CentraleSupelec notamment du Laboratoire Génie Industriel LGI ainsi que le personnel de l'école Doctorale INTERFACES et de l'INSTN du CEA pour mes trois années de scolarité. Ils ont effectué le suivi administratif de mon parcours. Ils m'ont également permis de suivre des formations, de faire des présentations vulgarisées ou des évaluations d'avancement des travaux.

Puis une pensée amicale aux équipes que j'ai pu côtoyer : le département LISE au CEA LIST (Nassim, François, Yupanqui, Benoit, Mathilde, Quentin, Gabriel, …), l'équipe AltaRica (Michel, Benjamin, Benoit, Mélissa, Loïc, Jean-Marc, Leïla, …). Ces personnes resteront gravées dans mon cœur, elles se sont personnellement investies dans mon travail particulièrement Michel Batteux. J'espère côtoyer toutes ces personnes encore longtemps.

Mes plus forts remerciements vont en premier lieu à mes parents, à ma sœur et à mes amies pour m'avoir accompagné et soutenu moralement. Ils ont généreusement contribués à la relecture de ce mémoire. Je tiens à remercier mon meilleur ami Baptiste pourles relectures de mes publications. Sa générosité, son écoute et son expertise dans les langues étrangères sont remarquables. Ma famille, mes amis ont été mon plus fort soutien, sans eux, je n'y serais jamais parvenu.

Je dédie ce mémoire à ma mère et mon père,
et à mes amis Baptiste K. et Aurélie H.

INTRODUCTION

1. OBJECTIFS DES TRAVAUX

Les systèmes produits par l'industrie deviennent de plus en plus complexes. Pour pallier à cette complexité, les différentes disciplines d'ingénierie, qui contribuent à leur conception (mécanique, hydraulique, thermique, électrique et électronique, software, hardware, middleware, …), virtualisent le contenu de leurs études au travers de modèles. Nous sommes entrés dans l'ère de l'ingénierie dirigée par les modèles [1]. De plus en plus utilisés, les modèles servent de support pour communiquer, pour calculer ou pour générer du logiciel embarqué.

Encore aujourd'hui, les disciplines d'ingénierie sont organisées en « silos ». Chacune utilise ses propres cadres mathématiques, concepts et méthodes. Les processus suivis sont spécifiques aux objectifs techniques et aux résultats attendus. Cette diversité se retrouve dans les modèles utilisés. Ces derniers sont conçus à l'aide de langages de modélisation différents (par exemple SysML [2], EAST-ADL [3], AUTOSAR [4], AADL [5], Simulink [1], Modelica [2], etc.), à des niveaux d'abstraction différents et avec des objectifs d'études différents. À une étape donnée du processus de conception, ils peuvent aussi présenter des niveaux de maturité très différents. En un mot, ils sont hétérogènes [6]. Cela peut amener les différents acteurs à avoir des visions divergentes, voire contradictoires du système en cours de conception.

Pour cette raison, les disciplines d'ingénierie interagissent tout au long du cycle de conception afin de s'accorder sur des spécifications intermédiaires ainsi que pour gérer les informations communes. Cependant, elles rencontrent de nombreuses difficultés dues à l'absence d'un vocabulaire commun, à l'ambiguïté des concepts utilisés et au manque de moyens techniques (cf. Annexe 1). De plus, dans l'état actuel des pratiques, ces interactions se placent quasi-exclusivement au niveau organisationnel. Les modèles sont peu ou pas concernés.

Cette thèse se propose d'étudier une solution potentielle à cette difficulté, via la notion de synchronisation de modèles [7].

*La **synchronisation de modèles** est un cadre théorique permettant d'établir une correspondance entre les contenus de modèles. Elle part du constat que les modèles ne peuvent pas être comparés directement dans leur formalisme respectif du fait de leur hétérogénéité. Elle propose donc d'abstraire le contenu des modèles dans un formalisme commun et de comparer ces abstractions. L'objectif de cette comparaison est d'assurer la cohérence des modèles, ou tout au moins, de permettre aux différentes disciplines d'ingénierie de se mettre d'accord sur leurs désaccords.*

Dans le cadre de cette thèse, nous nous sommes focalisés sur les interactions entre deux disciplines particulières : l'architecture système d'une part et la sûreté de fonctionnement d'autre part. Ces deux disciplines jouent, en effet, des rôles transverses dans le cycle de développement des systèmes. De plus, elles utilisent, toutes les deux, des modèles qui représentent la structure du système.

[1] The MathWorks, Inc.. [web1] "Simulink Tool", sur le site The MathWorks, Inc.. Consulté le 19 septembre 2017. https://fr.mathworks.com/products/simulink.html

[2] Modelica Association. [web2] « Modelica Language », sur le site Modelica Association. Consulté le 31 août 2017. https://www.modelica.org

L'hypothèse que nous avons cherché à tester peut s'énoncer de la façon suivante : « La synchronisation de modèles peut être employée comme moyen et support d'interaction entre l'architecture système et la sûreté de fonctionnement ».

Ce mémoire propose donc un cadre conceptuel et méthodologique ainsi que des outils logiciels pour appliquer une approche de synchronisation de modèles. L'objectif de l'approche développée est de construire et de maintenir la cohérence entre modèles hétérogènes tout au long du cycle de développement d'un système.

Tout au long de ce travail, nous avons cherché à rendre les concepts et les formalisations proposés suffisamment génériques pour envisager, plus tard, de les étendre aux interactions avec d'autres disciplines d'ingénierie.

2. LES DISCIPLINES D'INGENIERIE

La thèse se focalise sur l'architecture système et la sûreté de fonctionnement qui jouent, toutes les deux, un rôle majeur tout au long du cycle de développement des systèmes et qui ont de fortes dépendances avec les autres disciplines d'ingénierie.

L'architecture système définit, entre autres, la manière dont les composantes (fonctions, composants physiques, logiciels, etc.) du système sont architecturées, i.e. organisées et assemblées [8]. Elle s'appuie sur des représentations structurelles du système. Ces représentations caractérisent « l'architecture du système ». Pour les construire et les faire évoluer, l'architecture système interagit avec d'autres disciplines.

La sûreté de fonctionnement détermine et évalue les critères de fiabilité, de maintenabilité, de disponibilité et de sécurité du système en tenant compte de son architecture et du comportement de ses composantes [9]. Pour mener ces études, la sûreté de fonctionnement interagit avec l'architecture système et des disciplines techniques (mécaniques, électrique, pyrotechnique, etc.).

Ces deux disciplines d'ingénierie présentent une grande complémentarité. L'une propose une architecture du système et l'autre est chargée d'évaluer la sûreté de fonctionnement du système en tenant compte de la proposition d'architecture. Aujourd'hui, les interactions et les échanges entre ces deux disciplines sont informels. Ils s'effectuent au travers de réunions d'avant-projet ou de réunions de suivi, de lectures documentaires ou parfois de mécanismes de retranscription. Ces mécanismes s'effectuent généralement manuellement ou par le biais de passerelles (transformation de modèles [10]) spécifiques à un domaine industriel [11]. Ces interactions non formalisées posent de nombreux problèmes, compte tenu des risques d'incohérence entre les modèles d'un même système, dont les principaux sont :

- Le temps important et la duplication des efforts pour comprendre ce que le système doit faire et doit être ;
- Les difficultés de communication liées au vocabulaire utilisé pour désigner des concepts différents ou a contrario les mêmes concepts pouvant être désignés par des termes différents ;
- Les erreurs humaines dans la retranscription de l'information ;
- Les difficultés à retranscrire l'information contenue dans un modèle à un autre.

D'où l'intérêt de proposer de nouvelles approches collaboratives permettant d'améliorer les interactions entre ces deux disciplines d'ingénierie, tout en conservant une séparation des préoccupations.

3. LES MECANISMES DE SYNCHRONISATION DE MODELES

Les disciplines d'ingénierie ont besoin de collaborer pour échanger et se mettre d'accord sur leurs objectifs. Idéalement, leurs interactions devraient s'appuyer sur une comparaison des modèles qu'elles produisent. Cependant, cette comparaison est en pratique très difficile, voire impossible à réaliser, en raison de l'hétérogénéité des modèles. L'idée de la synchronisation de modèles est donc de mettre en œuvre des mécanismes permettant d'abstraire les modèles dans un formalisme commun, d'en étudier la cohérence à ce niveau d'abstraction, puis d'en restituer les résultats dans chaque modèle source. Cette approche est illustrée Figure 1.

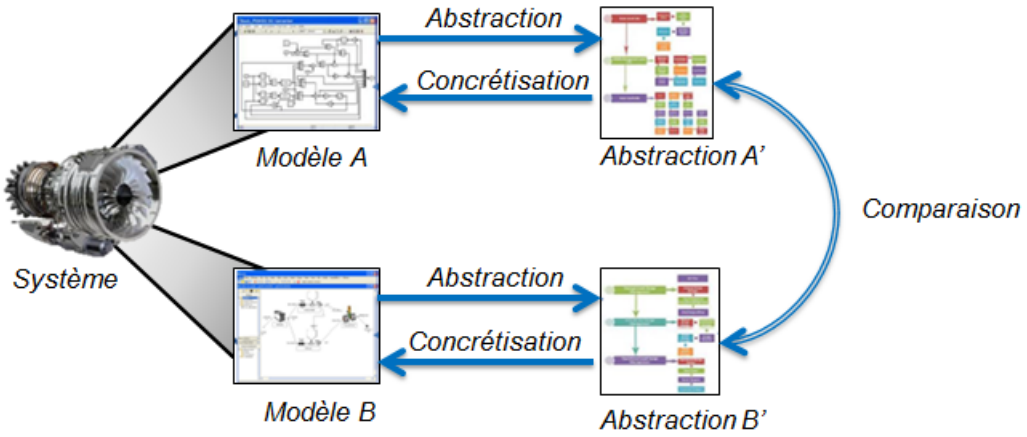


Figure 1 Principes de la synchronisation de modèles

La synchronisation de modèles est donc une démarche en trois étapes. Tout d'abord, une étape d'abstraction permettant d'extraire les informations pertinentes des modèles considérés. Ensuite une étape de comparaison des abstractions ainsi obtenues. Enfin une étape de concrétisation permettant de réinjecter des propositions d'aménagement issues des résultats de la comparaison vers les modèles originaux.

4. VERROUS SCIENTIFIQUES

L'approche proposée est à la fois pragmatique et originale. Elle demande de lever plusieurs verrous scientifiques et technologiques importants :

- Il faut appréhender le déroulement des processus de chaque discipline tout au long du cycle de développement du système et intégrer ces visions locales en une vision globale, dans le cadre particulier de l'entreprise ou du cas d'étude considéré.
- Il faut définir précisément les mécanismes de synchronisation de modèles dans le cadre particulier de l'entreprise ou du cas d'étude considéré.
- Il faut proposer une démarche de mise en œuvre de la synchronisation de modèles adaptée au contexte de l'entreprise ou du cas d'étude considéré.
- Il faut outiller cette démarche, ou en tout cas préciser ce que pourrait être à terme un outillage de cette démarche.

5. CONTRIBUTIONS

Afin de répondre à ces objectifs, le travail effectué apporte des contributions à plusieurs niveaux :

Contributions conceptuelles et méthodologiques :

- Caractérisation d'un besoin d'interactions multidisciplinaires
- Formalisation et proposition d'un cadre conceptuel pour la synchronisation de modèle
- Proposition d'une méthodologie pour appliquer la synchronisation de modèles en entreprise

Contributions relatives à l'applicabilité de l'approche sur un cas d'étude :

- Ces concepts et principes méthodologiques ont été mis en œuvre sur un cas d'étude représentatif

Contributions relatives à la faisabilité technique d'un outillage de la démarche :

- Réalisation de différents prototypes (abstraction, concrétisation)

Ces travaux ont donné lieu à plusieurs publications dans des conférences.

6. LISTE DES PUBLICATIONS

Trois publications scientifiques ont été publiées durant ma thèse.

LambdaMu20 : *Legendre, A., Lanusse, A., Rauzy, A.: Model synchronisation between architecture system and risk analysis: Which gain, how and why? In: CNRS (ed.) Conference: Congrès Lambda Mu 20 de Maîtrise des Risques et de Sûreté de Fonctionnement. LambdaMu20, IMdR, Saint Malo, France (Oct 2016), hal-01425284* – **Publié**

Résumé : *Nous avons observé un besoin de moyens collaboratifs entre les expertises métiers au plus tôt dans les processus, notamment entre la conception d'architectures système et les évaluations de la sûreté de fonctionnement. Dans cet article, nous introduisons des concepts et des outils conceptuels qui visent à mettre en cohérence entre les modèles et permettent un raffinement des modèles utilisés. Notre approche est illustrée ici sur un système embarqué dans un hélicoptère de combat.*

PSAM13 : *Legendre, A., Lanusse, A., Rauzy, A.: Directions towards supporting synergies between design and probabilistic safety assessment activities: illustration on a fire detection system embedded in a helicopter. In: PSAM13. Korean Nuclear Society, Seoul, South Korea (Oct 2016), hal-01425309* - **Publié**

Abstract: *The complexity of modern critical systems is growing rapidly while the industry is submitted to more and more pressure for reducing costs and time-to-market. Traditional development methods "in disciplinary silos" used to design and analyze such complex systems are reaching their limits. RAMS engineers face more and more difficulties to satisfy demands of reliability evaluation especially at early stages of system design. In this context we offer to take advantage of Model-Driven Engineering (MDE) approaches to reduce construction time of reliability models and improve their consistency with system models. Model-Driven Engineering is a promising approach used to develop and analyze complex systems from different domains. In this paper, we exploit MDE to support PSA analysis and illustrate the approach on a case study from avionics industry. This experimentation has enabled the suggestion of a seamless methodology to support iterative Probabilistic Safety Analysis, thus improving the cooperation with system designers.*

IMBSA2017 : *Legendre, A., Lanusse, A., Rauzy, A.: Toward model synchronization between safety analysis and system architecture design in industrial contexts. In: IMBSA2017. Trento, Italy (Sept 2017)* - **Publié**

Abstract: *Classical organization in disciplinary silos in the industry reaches its limits to manage complexity: problems are discovered too late and the lack of communication between experts prevents the early emergence of solutions. This is why it is urgent to provide new collaborative methods and ways to integrate various engineering fields, early in and all along the development cycle. In this context, we are particularly interested in the possible exchanges between two engineering fields: system architecture design and safety analysis. The questions are: how can one ensure that the parties involved are speaking about the same system? And which concepts can synchronize several engineering fields? First we present a use case: a system embedded in a helicopter. Second we present the concepts that we de ne to implement synchronization of models. Finally we give our feedbacks, limits and related works.*

7. PLAN DE LA THESE

Ce mémoire est organisé en 6 chapitres.

Le Chapitre I dresse un état de l'art scientifique en conception d'architecture système et en analyses de sûreté de fonctionnement. Il présente les missions, les méthodes, les normes, les outils, et les modèles utilisés. En complément, il aborde également l'ingénierie système et les problèmes liés aux interactions interdisciplinaires.

Le Chapitre II propose de définir le cadre conceptuel permettant d'appliquer une démarche globale de synchronisation de modèles. Ce cadre est composé de tâches formalisées de sorte que :

- Elles puissent se spécialiser pour tenir compte des besoins industriels ;
- Elles suivent les processus, les méthodes, les points de vue et les modèles utilisés tout au long du cycle de développement du système ;
- Elles configurent des interactions et permettent de valider la cohérence entre les modèles ;
- Elles mettent en œuvre les mécanismes de synchronisation de modèles.

Le Chapitre III propose une méthodologie d'application de la synchronisation de modèles. Elle s'appuie sur les concepts et les formalisations présentés au chapitre précédent. Pour chaque activité, les objectifs sont rappelés, la méthode est présentée et illustrée avec un ou plusieurs exemples.

Le Chapitre IV présente l'application de méthodes sur un cas d'étude issu du secteur industriel. Ces études sont le résultat des activités du processus d'architecture système et du processus de sûreté de fonctionnement. Elles sont effectuées de manière distincte, i.e. sans hypothèse d'interaction entre les disciplines.

Le Chapitre V présente le déroulement complet de la méthodologie de synchronisation (cf. Chapitre III) sur le même cas d'étude (cf. Chapitre IV). Au travers de ce déroulé, il présente cinq cas d'application de la synchronisation des modèles.

Le Chapitre VI termine ce mémoire. Il présente les travaux d'implémentation démontrant la faisabilité d'implémentation des techniques de synchronisation de modèles. Il montre notamment les moyens de :

- définir et utiliser des points de vue dans un environnement de modélisation ;
- mener des abstractions et des concrétisations à l'aide de transformations de modèles ;
- mener des comparaisons de modèles.

Le plan de la thèse et l'ensemble des contributions sont schématisés par la Figure 2.

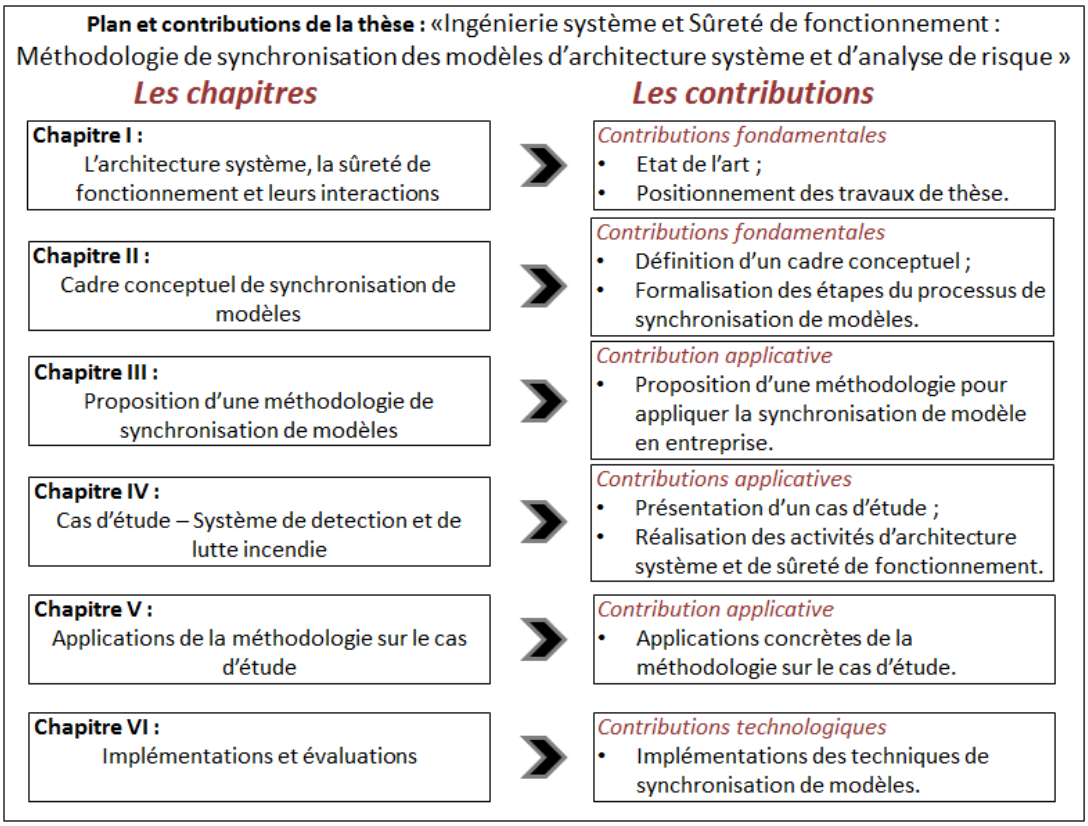


Figure 2 Contributions de la thèse sur la synchronisation de modèles

Chapitre I L'ARCHITECTURE SYSTEME, LA SURETE DE FONCTIONNEMENT ET LEURS INTERACTIONS

Avant d'aborder la synchronisation de modèles, nous nous intéressons au cadre qui englobe les activités des disciplines d'ingénierie, puis aux disciplines d'architecture système et de sûreté de fonctionnement. Nous terminons en énonçant les propositions récentes, basées sur les modèles, répondant aux problèmes d'interactions. Ce chapitre est donc découpé en quatre parties.

1. L'INGENIERIE SYSTEME

1.1. ORIGINES

Les institutions de la défense américaine : la NASA (National Aeronautics and Space Administration) et l'USAF (United States Air Force) ont été les premières à s'intéresser à l'ingénierie des systèmes [12]. Dans les années 60, elles ont tenté d'organiser le développement des programmes militaires et spatiaux (programme Apollo) à partir d'approches industrielles plus rationnelles.

Les années 70 et 80 voient une fulgurante avancée de l'informatique dans le pilotage des systèmes technologiques et parallèlement un net recul de l'ingénierie système [13]. Alors que le métier du logiciel s'organise et définit des méthodes du génie logiciel, la méthodologie du métier "système" stagne jusque dans les années 90.

Les constats d'échecs techniques ou économiques (l'inefficacité des systèmes de commandement durant les guerres, les pertes de satellites et l'explosion de navette spatiale) mettent en évidence des défauts d'origine systémique [12] (besoins peu exprimés, spécifications imprécises, solutions non justifiées ou non validées, confusion de responsabilités entre client et maîtrise d'œuvre …).

En 1991, l'International Council On System Engineering (INCOSE) est créé. Cette organisation capitalise et diffuse les activités intellectuelles et l'échange des bonnes pratiques pour le développement de systèmes complexes nécessitant l'interaction de plusieurs disciplines. Aujourd'hui, l'INCOSE poursuit ses actions et tente de répondre à de nombreuses problématiques autour de l'ingénierie des systèmes. Elle a notamment publié le document SE Handbook [14].

L'Association Française de l'Ingénierie Système (AFIS), fondée en 1998, travaille avec l'INCOSE pour diffuser les travaux d'ingénierie système en France et faire remonter également les résultats de groupes de travail au niveau national ou international. L'INCOSE et l'AFIS entretiennent la mise à jour du SE Handbook [14] et du SEBoK [15] (Systems Engineering Body of Knowledge).

1.2. PRESENTATION

Plusieurs communautés scientifiques, normatives et industrielles proposent des définitions différentes de l'ingénierie système (ou ingénierie de systèmes). C'est le cas de celles proposées par l'AFIS [16] et l'INCOSE [14]. Ces différences entrainent des confusions entre l'ingénierie système (le cadre des études) et la conception système (les activités réalisées par les disciplines d'ingénierie). Nous proposons donc la définition suivante :

L'ingénierie système *est un cadre qui englobe l'ensemble des activités des disciplines d'ingénierie. Ce cadre multidisciplinaire définit les périmètres des études des disciplines. Son but est de favoriser la réalisation d'un système performant comme solution finale aux besoins d'un client, tout en satisfaisant les* ***parties prenantes****.*

Une partie prenante *(« stakeholder ») est un acteur, individuel ou collectif (groupe ou organisation), activement ou passivement concerné par une décision ou un projet, i.e. dont les intérêts peuvent être affectés positivement ou négativement à la suite de son exécution (ou de sa non-exécution).*

L'architecture système et la sûreté de fonctionnement interviennent fortement dans les activités de conception, d'évolution et de vérification d'un système. Elles font partie du cadre d'ingénierie système.

Cependant, ni le cadre, ni les référentiels métiers des disciplines (présentés plus tard) ne définissent précisément avec qui, quoi et quand conduire les interactions entre ces disciplines. De plus, la sûreté de fonctionnement n'est pas toujours mentionnée dans les cours et les ouvrages d'ingénierie système [12].

Depuis plusieurs dizaines d'années, les différentes disciplines d'ingénierie conçoivent des modèles. Ces modèles doivent être pris au sérieux et considérés comme **l'objet central** des problématiques scientifiques des disciplines d'ingénieries.

1.3. NATURE ET FINALITE DES MODELES DE SYSTEME

La nature d'un modèle dépend du cadre utilisé : mathématique, informatique (langage, algorithme) ou graphique (représentation structurelle ou comportementale). Le concepteur de modèles choisit ce cadre en fonction du résultat recherché et des moyens possibles pour y parvenir.

Modèle (en ingénierie système) *[8] : C'est une abstraction d'un système réel ou étudié. Il s'appuie sur un cadre mathématique adapté à un ensemble d'objectifs et un ensemble de vues, i.e. des représentations définies par un point de vue.*

Un langage de modélisation (ou métamodèle) spécifie la sémantique et la ou les syntaxe(s) permettant de construire des modèles.

Syntaxe : *La syntaxe est, à l'origine, la branche de la linguistique qui étudie la façon dont les mots se combinent pour former des phrases ou des énoncés dans une langue.*
En informatique, la syntaxe définit des règles d'agencement des lexèmes (en informatique, ce sont des entités lexicales d'un langage informatique) en des termes plus complexes, souvent des programmes. Ces règles permettent de définir des expressions bien formées et de décider du respect, ou du non-respect, de la grammaire formelle d'un langage.

***Sémantique** : C'est une branche de la linguistique qui étudie les signifiés, ce dont on parle, ce que l'on veut énoncer. En informatique comme en linguistique, la sémantique désigne le lien entre un signifiant, le programme, et un signifié, l'objet mathématique qui dépendra des propriétés que l'on souhaite connaître du programme.*

La Figure 3 propose des exemples de modèles caractérisant le jeu du billard sous différentes préoccupations et à des niveaux d'abstraction très différents.

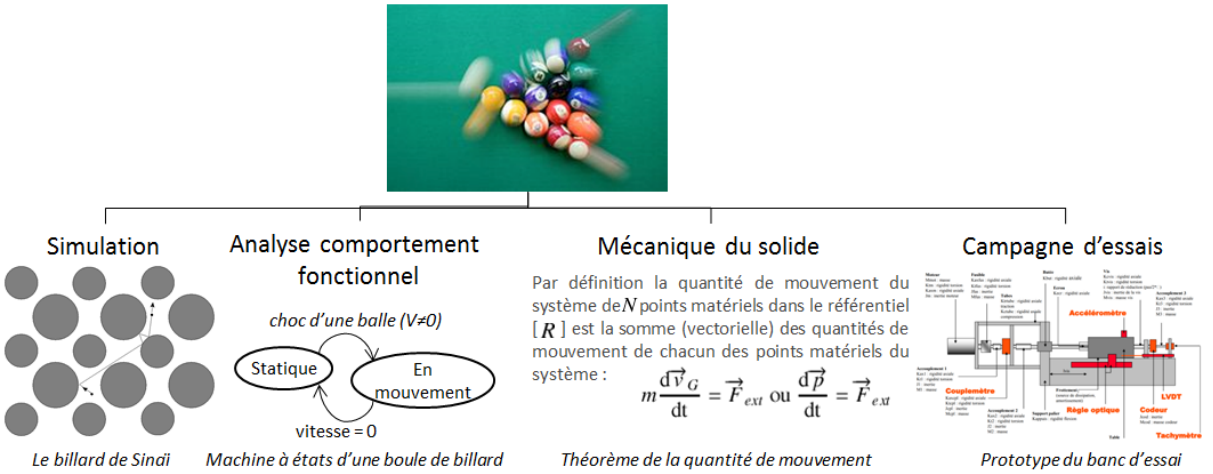


Figure 3 Quelques modèles étudiant le jeu du billard

En analysant les différents types de modèles qui permettent de représenter un système complexe, on peut constater qu'ils sont généralement conçus pour répondre à trois principaux objectifs :

- **Des modèles pour présenter.** Ils sont principalement employés pour construire des idées et communiquer avec des acteurs humains ayant des préoccupations particulières;
- **Des modèles pour calculer.** Ils sont principalement employés pour calculer des indicateurs ou simuler le comportement d'un système ;
- **Des modèles pour générer.** Ils sont employés pour construire une solution implémentée. Pour le logiciel, on parle de génération de code et pour des composants physiques, on parle de fabrication additive.

Aujourd'hui les « modèles pour présenter » sont décrits par des langages standardisés (génériques, cf.2.7.1 ou spécifiques, cf.2.7.2). Ils sont dits « semi-formels » car ils ne sont pas entièrement définis. Ils possèdent des points de variation sémantique [17], [18] qui permettent au concepteur de modèles de définir certaines caractéristiques du langage. Ces langages de modélisation ne sont pas suffisamment formels pour les deux autres objectifs.

Les « modèles pour calculer et pour générer » sont dit formels. D'une certaine façon, ils encodent et organisent des formules et des équations mathématiques. Ces modèles sont idéaux pour interagir avec des individus spécialistes et des machines de calculs.

Lorsqu'ils représentent un système complexe, les modèles sont eux-mêmes complexes. En effet, s'ils ont pour but d'abstraire une partie de la réalité du système, ils capturent nécessairement sa complexité sous des préoccupations particulières. Le cas contraire rendrait les modèles vains. Par conséquent, ils doivent être structurés, documentés, gérés ... Il font appel à l'**ingénierie des modèles**.

En architecture système, les modèles ont pour but de diffuser (présenter) des représentations de l'architecture du système.

En sûreté de fonctionnement, les modèles sont utilisés pour calculer, sur la base de données statistiques. La complexité des modèles et la précision des études requièrent alors l'utilisation de modèles formels.

L'objectif des modèles conditionne les langages de modélisation. L'hétérogénéité de ces modèles engendre des difficultés pour interagir entre disciplines et donc pour assurer la cohérence des études.

1.4. PROBLEMES MULTIDISCIPLINAIRES

Les interactions entre les disciplines d'ingénierie sont des opérations difficiles à mener [3]. Nous tentons d'en identifier les causes :

- la complexité croissante des systèmes ;
- l'organisation en « silos » des entreprises ;
- le nombre d'acteurs et de « strates » industriels dans les projets ;
- la demande accrue d'études plus amont dans les processus ;
- les problèmes de cohérence dans le discours.

La réalisation des systèmes industriels est de plus en plus complexe [19]. En effet, aujourd'hui, ils répondent à des services plus contraignants, spécifiques et contextualisés.

La structure des entreprises en « silo » [20] a permis de définir les processus, les méthodes et le cadre mathématique de chaque discipline d'ingénierie. Elle a permis l'approfondissement des connaissances techniques. Cependant, elle n'a pas tenu compte de leurs interactions. Cela peut engendrer des incompréhensions lorsque les disciplines sont amenées à interagir.

Les relations entre les entreprises sont également complexes. En effet, les « strates » constructeurs, systémiers, équipementiers et autres interviennent dans les projets de conception de systèmes. Les responsabilités ne sont pas clairement établies. Les échanges entre les « strates » d'entreprises s'appuient sur des livrables définis contractuellement. Les entreprises ont tendance à sous-traiter les activités dont elles n'ont pas la maitrise. Elles font appel à des sociétés de plus bas niveau dans la hiérarchie des systèmes plutôt que de développer ces compétences en interne, rendant le cycle de développement compliqué. Par exemple, il est difficile de retrouver les justifications des choix de conception. Ceci empêche toute anticipation d'écarts à un niveau global et favorise la sous-traitance des études non maitrisées ou potentiellement sensibles.

La sollicitation des ingénieurs de sûreté de fonctionnement pour l'évaluation de nouveaux systèmes ou fonctions critiques est de plus en plus importante. Les nouvelles versions des normes (prochaine version de l'ARP 4761 [21] ou encore l'ISO 26262 [22]) demandent, notamment, de rapides retours sur les évaluations dès les étapes amont du cycle de développement ; et ceci sans la garantie d'informations stables, ni de contexte précis.

Les échanges effectués par les ingénieurs entrainent souvent la transformation d'une représentation vers une autre. Souvent fait manuellement, ce travail chronophage est sujet à de

[3] Ledfrod, Heidi (2015). [web3] "How to solve the world's biggest problems", sur le site Nature. Consulté le 31 août 2017 https://www.nature.com/news/how-to-solve-the-world-s-biggest-problems-1.18367

nombreuses erreurs. Il peut introduire des écarts sémantiques du fait d'une mauvaise interprétation ou compréhension des modèles. Les mécanismes de traduction sont orientés et imposent une structure et un niveau d'abstraction aux ingénieurs qui les utilisent.

2. L'ARCHITECTURE SYSTEME

2.1. ORIGINES

A partir des années 90, l'architecture système est liée à l'architecture informatique [23] et à l'évolution des pensées systémiques [24]. L'architecture, au sens informatique, devient un terme incontournable à partir des années 80 et 90. Le terme "architecture" est considéré comme un buzzword et coïncide avec la multiplication des langages de programmation et l'essor de la micro-informatique.

Cette discipline, relativement « jeune », n'est pas entièrement mise en œuvre dans tous les secteurs industriels. Parfois, elle est associée au métier d'ingénieur système.

2.2. MISSIONS DE L'ARCHITECTURE SYSTEME

L'architecture système est une discipline centrée sur la construction de modèles représentant l'architecture d'un système. En effet, on ne saurait raisonner et plus généralement agir sur un système industriel, en cours de conception, sans décrire son organisation interne et l'intégration de ses composants [12].

La méthode CESAM [12], l'AFIS [25] et l'INCOSE définissent trois principales visions de l'architecture système :

- **Une vision opérationnelle** ; Elle a pour but de définir le *pourquoi* du système en décrivant son environnement.
- **Une vision fonctionnelle** ; Elle permet d'identifier les fonctions d'un système et d'expliquer leur organisation (fonctionnement logique), indépendamment de la façon dont elles seront réalisées.
- **Une vision physique** *(parfois appelée vision organique)* ; Elle définit la façon dont le système est concrètement réalisé, i.e. l'organisation et la dynamique de ses composants matériels, logiciels et humains.

La Figure 4 synthétise les visions architecturales d'un système [25] selon l'AFIS.

L'élaboration de ces visions constitue la mission principale de l'architecture système. Elle induit une seconde mission, celle de devoir interagir avec les autres disciplines d'ingénierie pour obtenir des retours d'analyse et d'évaluation de ces visions. L'architecte système (ingénieur de l'architecture système) doit adapter ses représentations (modèles 3D, diagrammes, algorithmes, documents, maquettes, animations, ...) selon la discipline avec laquelle il traite.

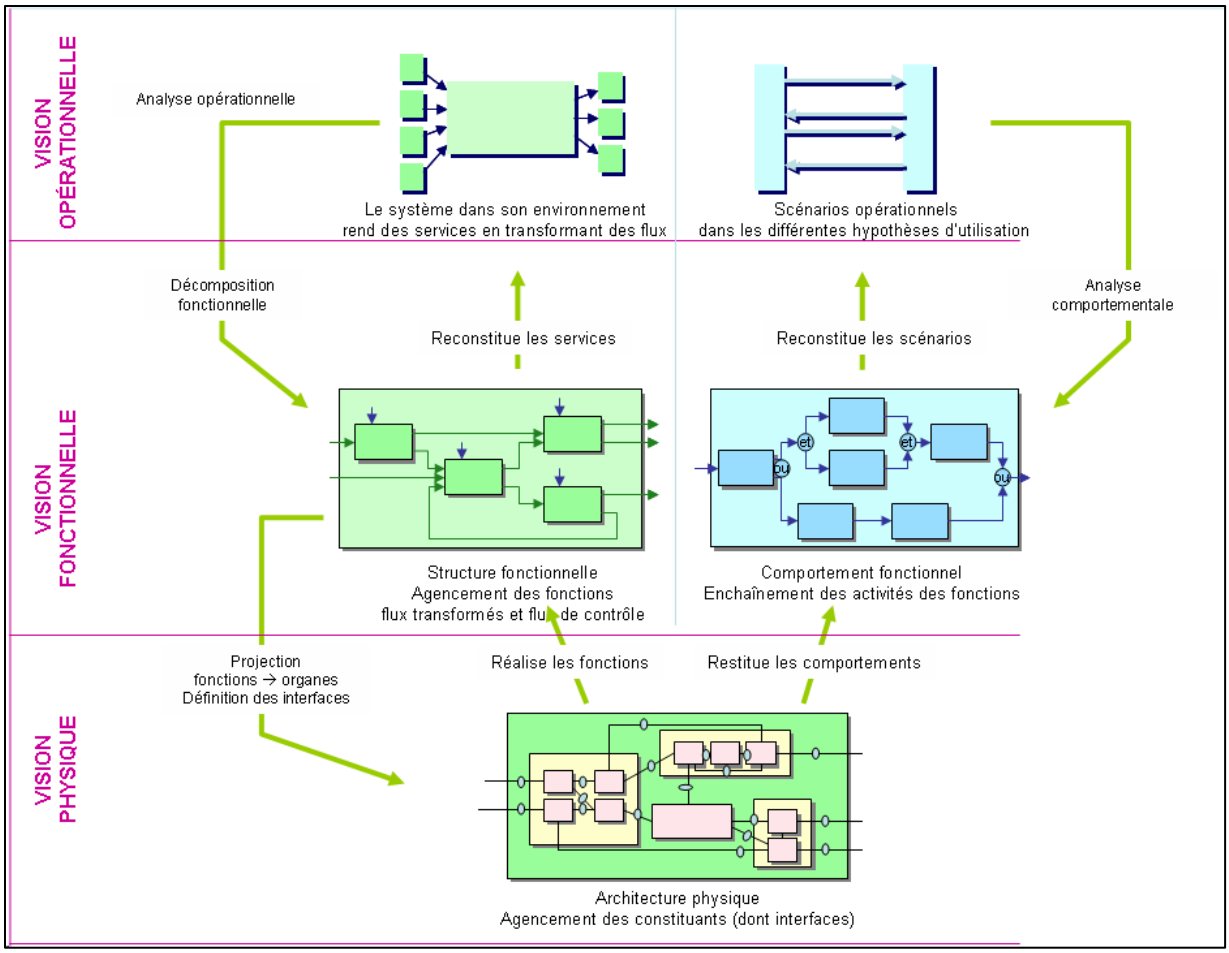


Figure 4 Les principales visions architecturales d'un système technologique

L'architecture système se définit donc ainsi :

L'architecture système : *Elle est responsable de deux missions.*
La première est de proposer des solutions d'architecture, i.e. définir comment les composantes (fonctions, composants physiques, logiciels, etc.) du système sont organisées et assemblées entre elles. Ces solutions sont contenues dans des modèles à plusieurs niveaux d'abstraction et de raffinement (vision opérationnelle, vision fonctionnelle, vision organique plus ou moins détaillées).
La seconde mission est d'interagir avec les disciplines techniques et transverses, pour obtenir des retours sur les solutions d'architecture proposées. Elle doit adapter ses représentations selon les préoccupations des disciplines ciblées.

L'architecte système peut faire de meilleurs compromis sur la base des recommandations obtenues. L'architecte a donc un rôle de médiateur entre les disciplines d'ingénierie. De plus, il pondère les retours obtenus pour optimiser les choix de conception et faire évoluer la solution d'architecture.

2.3. TAXONOMIE ET PARADIGME DE STRUCTURATION

Aujourd'hui, il n'y a pas de taxonomie établie en architecture système. Les communautés proposent des langages et des normes qui induisent implicitement des paradigmes de structuration. Un paradigme de structuration met en avant comment le contenu d'un modèle peut représenter l'architecture d'un système.

Plusieurs organismes proposent des langages de modélisation et des standards pour tenter de formaliser l'activité de modélisation. Ces travaux varient selon :

- le type d'objet représenté (système, application, logiciel, 3D, etc.) ;
- le champ d'application (global, par secteur industriel, etc.) ;
- la raison qui a initié ce travail et le choix de la démarche.

Les principaux organismes sont :

- L'Object Management Group (OMG, association) a pour objectif de standardiser et de promouvoir les modèles sous toutes ses formes. Elle propose des langages très génériques (UML [26], Unified Modeling Langage, SysML [2], System Modeling Langage, BPMN [27], Business Process Model and Notation). Ils se spécialisent pour représenter le plus d'aspects possibles.
- La SAE International (anciennement Society of Automotive Engineers) échange des informations et des idées sur l'ingénierie des véhicules au sens large (automobile, aéronautique, etc.). Elle propose le langage AADL « Architecture Analysis and Design Language » [5]. Ce langage a été défini pour représenter les architectures de systèmes aéronautique et spatial.
- De la même manière, les projets européens ATTEST puis MAENAD [28] ont défini le langage EAST-ADL [3], [29] pour la description d'architectures de systèmes embarqués automobile.
- Enfin, l'organisme international de normalisation (ISO) définit des standards pour homogénéiser les concepts liés à l'architecture système (cf. 2.5 de ce chapitre).

2.4. METHODES

Les méthodes employées pour concevoir des architectures de système sont fondées sur les méthodes d'analyses fonctionnelles telles que SADT [4], SA/RT[5], etc.

La virtualisation des contenus a fait évoluer les méthodes d'analyse fonctionnelle en méthodes de modélisation d'architecture. Les principes de la modélisation orientée objet [30] ont permis d'offrir de nouvelles techniques de manipulation et de gestion des contenus de modèles. Ils permettent une meilleure représentation des préoccupations [8]. Ce principe s'appelle : model-based system engineering, il permet de travailler et de communiquer directtement sur des modèles en opposition aux démarches plus classiques dites « document-based » où les échanges se font sur la base de documents produits puis transmis.

L'INCOSE définit [14] : « L'ingénierie système basée sur les modèles (MBSE), est l'application formalisée de la modélisation permettant la gestion des exigences du système, la conception, l'analyse, les activités de vérification et de validation dès les étapes amont et tout au long du cycle de vie ».

A nouveau, l'INCOSE est imprécise dans sa définition, elle induit des erreurs entre l'ingénierie système et la discipline d'architecture système. Pour le besoin de notre étude, nous allons restreindre la définition de l'INCOSE à la suivante :

[4] Philippe Berger. [web4] « Analyse S.A.D.T. », sur le site Page personnelle de Philippe Berger sur free.fr. Consulté le 31 août 2017. http://philippe.berger2.free.fr/automatique/cours/sadt/sadt.htm#top

[5] Case France. [web5] « Analyse et conception SART », sur le site Case France. Consulté le 31 août 2017. http://www.case-france.com/EnvisionSART.html

*« **L'ingénierie système basée sur les modèles (MBSE)**, est l'application formalisée de la modélisation permettant à l'architecture système de gérer ses modèles et ses activités tout au long de son processus. »*

2.5. NORMES

Deux familles de normes concernent les activités de conception d'architecture système. Une première famille est dédiée à l'architecture système. Ces normes sont relatives à la définition, à l'élaboration et à l'évaluation d'une architecture d'un système.

- ISO 42010: Systems and software engineering — Architecture description [8] ;
- ISO 42020: Systems and software engineering — Architecture processes [31] (en cours de développement) ;
- ISO 42030: Systems and software engineering — Architecture evaluation [32] (en cours de développement).

Une seconde famille de normes traite du processus appliqué durant le cycle de développement d'un système. Elle impacte les disciplines du cadre d'ingénierie système, notamment l'architecture système.

- IEEE 1220 : IEEE Standard for Application and Management of the Systems Engineering Process [33] ;
- EIA 632: Processes for Engineering a System [34] ;
- ISO 15288: Systems and software engineering — System life cycle processes [35].

2.6. LES LOGICIELS

Des universitaires [36], [10] ont mené un benchmark détaillé des langages de modélisation d'architecture. Ce travail liste les langages et les outils permettant de représenter des architectures au sens large (pas nécessairement d'un système). D'autres travaux proposent une comparaison des outils de modélisation basés sur le langage UML [6].

Nous avons listé les principaux outils de modélisation permettant de mener des activités d'architecture système. La Table 1 compare ces logiciels selon les éléments de comparaison suivants : nom de l'éditeur, le type d'environnement, les fonctionnalités proposées, les langages sur lesquels s'appuient les outils.

Table 1 Comparaison des logiciels de modélisation

Nom	Editeur	Environnement	Fonctionnalités	Langage	License
Cameo Systems Modeler [7]	No Magic	Environnement de modélisation MBSE basé sur SysML	Modélisation d'architecture système, Analyses, Génération de documents, Résolution de contraintes	Magic Draw avec un plugin SysML	Payante

[6] Wikimedia Foundation (2010). [web6] « Comparaison des logiciels UML », sur le site Fracademic. Consulté le 31 août 2017. http://fracademic.com/dic.nsf/frwiki/1888634

[7] No Magic. [web7] "Cameo Systems Modeler Tool ", sur le site No Magic. Consulté le 31 août 2017. https://www.nomagic.com/products/cameo-systems-modeler#features

Nom	Editeur	Environnement	Fonctionnalités	Langage	License
Enterprise Architect [8]	Sparx Systems	Environnement de modélisation MBSE basé sur UML	Modélisation d'architecture système	UML, SysML, BPMN	Payante
Modelio SA [9]	ModelioSoft	Environnement de modélisation MBSE basé sur SysML, UML/BPMN	Modélisation d'architecture système, Analyses, Génération de document, Modélisation d'architecture d'entreprise.	SysML, UML et BPMN, NAF/DoDAF/ MODAF	Payante
IBM Rhapsody [10]	IBM	Environnement de modélisation MBSE basé sur UML, SysML ou des DSL (AUTOSAR)	Modélisation d'architecture système et d'exigences	SysML, UML	Payante
ARTiSAN Studio [11]	PTC (anciennement Atego)	Environnement de modélisation MBSE basé sur SysML	Modélisation d'architecture système	UML, SysML	Payante
Capella [12]	Projet PolarSys, OBEO	Environnement de modélisation MBSE et de simulation basé sur UML	Modélisation d'architecture système, Méthodologie Arcadia [37]	UML, SysML, BPMN	Open-source
Papyrus [13]	Projet PolarSys, CEA LIST	Environnement de modélisation MBSE et de simulation basé sur UML, SysML et MARTE	Modélisation d'architecture systèmes	UML, SysML, BPMN, MARTE	Open-source

Les logiciels se différencient par les langages de modélisation implémentés et les vues mises à disposition de l'utilisateur. Pour aller plus loin, il faut s'intéresser plus en détail aux modèles et aux langages qui les implémentent.

2.7. LES LANGAGES DE MODELISATION

2.7.1. LES LANGAGES GENERIQUES

Même si les langages « génériques » ont des objectifs bien plus ambitieux que de modéliser l'architecture d'un système, dans le cadre de cette thèse, nous les considérerons comme tels.

L'OMG, est l'organisme principal qui propose ce type de langages. Elle définit notamment :

- UML [26] comme un standard de modélisation logicielle ;
- SysML [2] comme un standard de modélisation système ;
- SPEM [38] et BPMN [27] comme des standards de modélisation de processus.

[8] Sparx Systems. [web8] "Enterprise Architect Tool ", sur le site Sparx Systems. Consulté le 31 août 2017. https://www.sparxsystems.com/products/mdg/tech/sysml/index.html

[9] ModelioSoft. [web9] « Modelio SA Tool », sur le site ModelioSoft. Consulté le 31 août 2017. https://www.modeliosoft.com/en/products/modelio-sa-overview.html

[10] IBM. [web10] "IBM Rhapsody Tool", sur le site IBM. Consulté le 31 août 2017. http://www-03.ibm.com/software/products/en/ratirhaparchforsystengi

[11] PTC. [web11] « ARTiSAN Studio Tool », sur le site PTC. Consulté le 31 août 2017. https://www.ptc.com/en/model-based-systems-engineering/integrity-modeler

[12] OBEO. [web12] « Capella Tool », sur le site Polarsys. Consulté le 31 août 2017. https://www.polarsys.org/capella

[13] CEA LIST. [web13] "Papyrus Tool", sur le site Polarsys. Consulté le 31 août 2017. https://www.polarsys.org/solutions/papyrus

Ces langages, UML en particulier, sont extensibles, grâce à un profil. Ce dernier est un mécanisme d'extension du langage. Il permet d'ajouter des stéréotypes et des propriétés, liés à des concepts du langage, sans contradiction avec la sémantique préexistante. Notons qu'il n'existe pas de mécanisme inverse, i.e. un mécanisme de restriction d'un langage. Actuellement, les solutions proposées sont d'ajouter une surcouche au modèle avec un langage de contraintes comme OCL [39].

SysML (OMG-SysML, 2006 [40]) est un langage dédié à la modélisation de système. C'est un profil d'UML, plus particulièrement une extension d'un sous-ensemble d'UML [26]. Il est aussi possible de définir des profils pour étendre SysML.

Ces langages sont si abstraits qu'ils sont difficiles à rattacher à la réalité. Comme SysML n'est pas spécifique à une activité ou une discipline, la connaissance technique et le contexte d'étude sont à construire par le concepteur de modèle. C'est un exercice intellectuel difficile.

2.7.2. LES LANGAGES SPECIALISES

Les DSML ont été introduits pour construire des représentations de système dans un contexte industriel particulier.

***Un langage de modélisation spécifique au secteur d'activité (DSML)** : C'est un langage de modélisation (généralement graphique) pour créer des modèles spécifiques à un certain domaine (e.g. diagnostic de maladie, configuration « Quality of Service »). Il offre une expressivité axée sur une problématique particulière grâce à une notation et des abstractions appropriées.*

Les DSML les plus répandus sont les suivants :

AADL [5]***:*** Le langage d'analyse et de conception d'architecture (AADL) est une norme SAE spécialement centrée sur la conception et l'analyse d'architecture logicielle et matérielle des systèmes performants. Il tient compte des aspects temps réel. De plus, il est utilisé pour décrire la structure des systèmes comme un assemblage de composants logiciels mappés sur une plateforme d'exécution. Il peut décrire les interfaces fonctionnelles jusqu'aux composants (telles que les interfaces d'entrées et de sorties de données) et tenir compte des performances critiques des composants. Il décrit également comment les composants interagissent, comme les interconnexions entre les entrées et les sorties de données et les allocations des composants logiciels aux composants de la plateforme.

AADL [5], comme SysML, dispose d'un mécanisme d'extension qui permet de tenir compte de nouveaux objectifs d'étude et de propriétés matérielles spécifiques. Il utilise un modèle « Error Annex » qui peut être associé à des composants.

EAST-ADL [41, 3]***:*** c'est un langage de modélisation d'architecture de systèmes embarqués dans le secteur automobile. Développé dans le cadre du projet EAST-EEA, il propose de nombreuses vues depuis les premières étapes du cycle de développement jusqu'à l'implémentation d'une solution. Ces vues sont définies à 7 niveaux d'abstraction différents, chacune couplée avec des exigences : *Vehicule View, Functional Analysis Architecture, Functional Design Architecture, Function Instance Model*, *Hardware Architecture*, *Platform Model, Allocation Model.*

On retrouve des chevauchements entre les architectures, ce processus peut être adapté aux besoins des disciplines et aux entreprises. Ce langage propose un cadre de modélisation avec des vues cohérentes.

CAPELLA est un outil d'architecture système associé à la méthode Arcadia [37]. Il couvre les activités de définitions, d'analyses et de validations de systèmes. La méthode Arcadia [37] possède les caractéristiques suivantes :

- Les modèles manipulés prennent en charge la collaboration et la co-ingénierie à l'échelle de l'entreprise en partageant une architecture de référence pour toutes les analyses menées ;
- La méthode propose des étapes adaptées à la conception d'architecture, de l'analyse du besoin au développement ;
- Capella propose d'utiliser un DSL (Domain Specific Language) approprié pour les utilisateurs débutants, non familiers avec les langages génériques.

3. ANALYSES DE SURETE DE FONCTIONNEMENT

3.1. HISTORIQUE

La revue InterSections de novembre 2004 [42] présente les évolutions des études d'analyse de risques et de sûreté de fonctionnement. Pour enrichir cette étude, nous avons construit une frise chronologique qui recueille des évènements accidentels et les explorations de nouvelles démarches en sûreté de fonctionnement[14]. Une soixantaine d'évènements ou périodes de l'histoire, entre les années 1600 à 2006, ont été capitalisés et listés dans la Table 2.

Table 2 Références historiques de sûreté de fonctionnement

Date début	Type d'évènement	Titre	Références
1600	Approche	Etude du maillon le plus faible d'une chaîne de production (261 ans) – Les études se limitaient à l'analyse du poste de travail le plus lent d'une ligne de fabrication.	[42]
1794	Accident	Explosion de la poudrière de Grenelle	[15]
1842	Accident	Accident de chemin de fer de Meudon	[16]
1861	Evènement	Conquête de l'ouest 1861-1930 (69 ans) - Les composants mécaniques les plus critiques de l'époque étaient les roulements à billes des locomotives à vapeur.	[42]
1912	Accident	Titanic (2471 Morts)	[17]
1914	Evènement	Première Guerre Mondiale 1914-1918 (4 ans) - les bateaux construits rapidement pour amener les soldats américains sur le sol européen ne résisteront que très difficilement aux eaux gelées de l'Atlantique Nord.	[42]
Années 1930	Approche	1er objectif de Safety, par le capitaine A.F. Pugsley de la 7ème brigade d'infanterie canadienne	[42]
Années 1930	Approche	1er Approche statistique, Taux de défaillance	[42]
Années 1930	Approche	1er estimation de la probabilité d'accident d'un avion. « Avec un taux de défaillance évalué à 10^{-5}/h pour les avions, dont 10^{-7}/h pour leur structure ».	[42]

[14] Anthony Legendre. [web14] "Frise chronologique des évolutions des études en sûreté de fonctionnement", sur le site Timeglider. Consulté le 22 septembre 2015. http://fresques.ina.fr/jalons/fiche-media/InaEdu01039/le-naufrage-du-petrolier-torrey-canyon.html

[15] [web15] "Chronologie de catastrophes industrielles ", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Chronologie_de_catastrophes_industrielles

[16] [web16] "Liste des accidents ferroviaires en France au xixe siècle", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Liste_des_accidents_ferroviaires_en_France_au_XIXe_si%C3%A8cle

[17] [web17] "Naufrage du Titanic", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Naufrage_du_Titanic

Date début	Type d'évènement	Titre	Références
Années 1940	Approche	Apparition de la Fiabilité Prévisionnelle	[42]
Années 1940	Approche	1er Qualification de la disponibilité	[42]
Années 1940	Approche	Loi de Murphy « Tout ce qui est susceptible de mal tourner tournera nécessairement mal. » — Edward A. Murphy Jr.	[42]
Années 1940	Evènement	Analyse des missiles allemands V1, W. Von Braun tente de prouver que la fiabilité d'une chaîne est la moyenne de la fiabilité de ses constituants	[42]
1948	Evènement	Projet MX 981 : Essais américains en Avionique	[42], [43]
Années 1950	Approche	1er recueil de Fiabilité par le Centre National d'Etudes sur les Télécommunications de France.	[42]
Années 1950	Approche	1er étude sur le coût de la maintenance : « On assiste à l'avènement du concept de maintenance : 1 dollar en équipement génère 2 dollars en maintenance. »	[42]
Années 1950	Evènement	La Marine américaine prend conscience des taux d'utilisation exécrables de ses bâtiments.	[42]
1959	Accident	Effondrement du barrage de Malpasset (423 Morts)	[44]
Années 1960	Approche	1er version de la MIL-HDBK-217	[45]
Années 1960	Approche	1er analyse de la défaillance (Aéronautique + Spatial) - Programme APOLLO	[42]
Années 1960	Approche	1er Arbre de causes (Boeing - NASA)	[42]
Années 1960	Approche	1er design des arbres de défaillances	[18]
Années 1960	Approche	Début des analyses des modes de défaillance et de leurs effets	[42]
1963	Accident	Eboulement de terrain au lac artificiel Vajont (2168 Morts)	[19]
1965	Approche	Début de l'harmonisation et de la normalisation internationale des études de Sûreté de Fonctionnement (CEI)	[42]
1965	Evènement	Introduction des concepts de maintenance au CEA	[42]
1967	Accident	Accident de l'USS Forrestal	[20]
1967	Accident	Pétrolier Canyon - 123 000T de pétrole déversées sur les côtes (GB+FR)	[21]
Années 1970	Approche	EDF & CEA - 1er étude exhaustive sur les centrales nucléaires françaises	[42]
Années 1970	Approche	1er résultat sur la fiabilité logicielle	[42]
Années 1970	Evènement	Publication du Rapport Rasmussen	[46], [42]
1975	Accident	Rupture en cascade de 62 barrages en Chine (26000 Morts puis 145000 Morts de maladie)	[22]
1979	Accident	Rupture du Barrage Morvi (15 000 Morts)	[23]
1979	Accident	Three Miles Island - Fusion partielle du cœur	[24], [42]

[18] [web18] "Arbre de défaillances Historique", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Arbre_de_d%C3%A9faillances#Historique

[19] [web19] "Barrage de Vajont", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Barrage_de_Vajont

[20] [web20] "Accident de l'USS Forrestal ", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Accident_de_l%27USS_Forrestal

[21] [web21] INA. "Le naufrage du pétrolier Torrey Canyon", sur le site Jalons Version découverte. Consulté le 31 août2017. http://fresques.ina.fr/jalons/fiche-media/InaEdu01039/le-naufrage-du-petrolier-torrey-canyon.html

[22] [web22] "Barrage de Banqiao", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Barrage_de_Banqiao

[23] [web23] "Catastrophe de Morvi", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Catastrophe_de_Morvi

[24] [web24] "Accident nucléaire de Three Mile Island ", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Accident_nucl%C3%A9aire_de_Three_Mile_Island

Date début	Type d'évènement	Titre	Références
Années 1980	Approche	Augmentation des campagnes d'essais nucléaire français	[25]
Années 1980	Approche	Efforts de Normalisation	[42]
Années 1980	Approche	Mise en place des Cercles Q (Qualité) en Europe	[26]
1984	Accident	Terminal PEMEX de gaz et pétrole liquéfié (1700 Morts)	[47]
1986	Accident	Fusion du cœur - Tchernobyl	[27]
1986	Accident	Destruction de la navette Challenger	[28]
1999	Evènement	Première version de l'atelier AltaRica	[48]
2005	Evènement	UTE C 80 810	[49]
2006	Evènement	Création de l'IMDR (Institut pour la Maîtrise de Risques)	[50]

L'évolution des approches de sûreté de fonctionnement est liée aux évènements et accidents de l'histoire.

3.2. MISSIONS DES ETUDES DE SURETE DE FONCTIONNEMENT

Alain Villemeur est l'un des pionniers des analyses de risques (discipline englobant la sûreté de fonctionnement). Il définit dans son livre [9] l'activité de sûreté de fonctionnement :

La sûreté de fonctionnement *(dependability, SdF) consiste à évaluer les risques potentiels, prévoir l'occurrence des défaillances et tenter de minimiser les conséquences des situations catastrophiques lorsqu'elles se présentent.*

Pour aller plus loin, nous proposons une définition complémentaire. Elle s'inspire des définitions normatives [51] :

La sûreté de fonctionnement *est l'ensemble des aptitudes d'un système à remplir une fonction requise au moment voulu, pendant la durée prévue, sans dommage pour lui-même et son environnement.*

La sûreté de fonctionnement est une discipline qui définit et évalue les niveaux de risques associés aux évènements non désirés du système. L'ingénieur évalue le comportement dysfonctionnel du système en identifiant les défaillances, leurs causes, leurs effets sur lui-même et son environnement. Il se base sur des essais, le retour d'expérience et sur l'avis d'experts. Il capture des scénarios d'échecs ou d'accidents qui font évoluer l'état du système. Les analyses et les évaluations s'appuient sur des évènements discrets [52] basés sur des statistiques du comportement des composants.

Les activités de l'ingénieur de sûreté de fonctionnement [51] englobent les analyses et les évaluations des attributs de fiabilité, de maintenabilité, de disponibilité et de sécurité. L'ingénieur étudie la complétude entre les performances calculées ou estimées et les exigences du système. L'objectif étant de vérifier l'acceptabilité du risque décidé en amont du projet.

[25] [web25] "Essais nucléaires français ", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Essais_nucl%C3%A9aires_fran%C3%A7ais

[26] [web26] "Cercle de qualité", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Cercle_de_qualit%C3%A9

[27] [web27] "Catastrophe nucléaire de Tchernobyl", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Catastrophe_nucl%C3%A9aire_de_Tchernobyl

[28] [web28] "Accident de la navette spatiale Challenger", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Accident_de_la_navette_spatiale_Challenger

3.3. TAXONOMIE

Des efforts considérables ont été faits par Algirdas Avizienis et Jean-Claude Laprie [53], [54] pour définir une taxonomie de la sûreté de fonctionnement. Ils organisent la sûreté de fonctionnement en trois concepts :

- Les attributs : propriétés quantifiables, évaluables, caractérisant les performances du système ;
- Les entraves : évènements qui peuvent affecter les performances du système ;
- Les moyens : techniques pour améliorer les valeurs des attributs.

La déclinaison de ces concepts est appelée « Arbre de la sûreté de fonctionnement » (Figure 5).

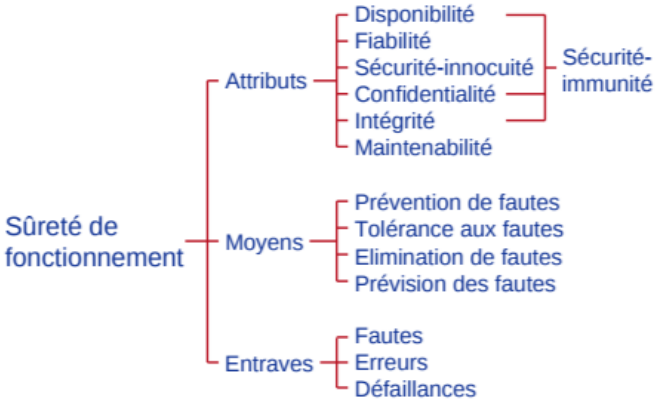


Figure 5 Arbre de la sûreté de fonctionnement

Les attributs de la sûreté de fonctionnement [51] se caractérisent selon l'ouvrage de Villemeur [9], par :

- Fiabilité : Aptitude d'une entité S à accomplir une fonction requise, dans des conditions données pendant un intervalle de temps donné.

$$\boldsymbol{R(t) = prob(S\ fonctionne\ sur\ [0, t]) \in [0, 1], avec\ t, une\ date, \in \mathbb{R}^+}$$

- Disponibilité : Aptitude d'une entité à être en état d'accomplir une fonction requise dans des conditions données, à un instant donné (ou pendant un intervalle de temps donné), en supposant que la fourniture des moyens extérieurs nécessaires soit assurée.

$$\boldsymbol{A(t) = prob(S\ fonctionne\ à\ l'instant\ t), avec\ t, une\ date, \in \mathbb{R}^+}$$

- Maintenabilité : Aptitude d'une entité à être maintenue ou rétablie, dans un intervalle de temps donné dans un état dans lequel elle peut accomplir une fonction requise, lorsque la maintenance est accomplie dans des conditions données avec des procédures et des moyens prescrits.

$$\boldsymbol{M(t) = prob(S\ réparée\ sur\ [0, t]) \in [0, 1], avec\ t, une\ date, \in \mathbb{R}^+}$$

- Sécurité innocuité : Aptitude d'un produit à respecter, pendant toutes les étapes du cycle de vie, un niveau acceptable de risques susceptibles d'occasionner une agression du personnel, une dégradation majeure du produit ou de son environnement.

$$\boldsymbol{S(t) = prob(S\ génère\ aucun\ risque\ sur\ [0, t]) \in [0, 1], avec\ t, une\ date, \in \mathbb{R}^+}$$

3.4. METHODES

Les méthodes appliquées en sûreté de fonctionnement permettent de caractériser le comportement dysfonctionnel d'un système. Les formalismes utilisés sont spécifiques et ont été développés pour des objectifs d'études très particuliers.

De nombreuses méthodes [55] utilisées répondent à des problématiques de sûreté. Elles sont qualitatives lorsqu'elles caractérisent la nature des risques associés aux éléments du système. A l'inverse, elles sont quantitatives lorsqu'elles mesurent les attributs de la sûreté de fonctionnement. Généralement, les méthodes qualitatives précèdent les méthodes quantitatives. Les méthodes peuvent être caractérisées d'inductives ou de déductives [56].

Nous proposons dans la Table 3 un classement des principales méthodes d'analyses de risques et de sûreté de fonctionnement :

Table 3 Familles de méthodes utilisées dans le cadre d'études en sûreté de fonctionnement

Famille de méthode	**Méthodes**	**Inductive/déductive + Qualitative/quantitative**
Retour d'expérience	Retour d'expérience----------	Déductive, qualitative et quantitative
Analyse fonctionnelle	MISME [57]---------- SADT (ou IDEF0) ---------- FAST [29] ---------- MERISE[30] ---------- SA/RT [58]----------	Déductive et qualitative Déductive et qualitative Déductive et qualitative Déductive et qualitative Déductive et qualitative
Analyse de risques fonctionnels	PHA (Analyse préliminaire de risques) [59]---------- SHA (Analyse de risques système) ---------- FHA (Analyse de risques fonctionnels) [21]---------- HAZOP [60]---------- MOSAR [61]----------	Inductive et qualitative Inductive et qualitative Inductive et qualitative Inductive et qualitative Inductive et qualitative
Analyse des modes de défaillances	AMDE (FMEA) [62]---------- AMDEC (FMECA) [63]---------- AEEL [64]----------	Inductive et qualitative Inductive et quantitative Inductive et qualitative
Analyses sur des formalismes booléens	Arbre de causes [65]---------- Arbre d'évènement [66]---------- Arbre de défaillance [67]----------	Déductive et quantitative Inductive et quantitative Déductive et quantitative
Analyses avec des systèmes de transition	Réseau de PETRI [68]---------- Chaine de Markov [69]---------- Système de transition gardée [70]----------	Inductive et quantitative Inductive et quantitative Inductive et quantitative

L'essentiel des méthodes de la Table 3 sont formelles. La Figure 6 représente les formalismes graphiques employés par les méthodes quantitatives. Elles se catégorisent en deux types : les formalismes booléens et les systèmes de transition.

[29] [web29] AV Canada (2015). "LA MÉTHODE FAST (FONCTION ANALYSIS SYSTEM TECHNIQUE)", sur le site AV Canada. Consulté le 31 août 2017. http://scav-csva.org/fast.php?lang=fr

[30] [web30] ESPINASSE, Bernard. "MERISE : une méthode systémique de conception de SI", sur le site Laboratoire des sciences de l'information et des systèmes. Consulté le 31 août 2017. http://www.lsis.org/dea/M6optionD/Exp-GL5Merise.pdf

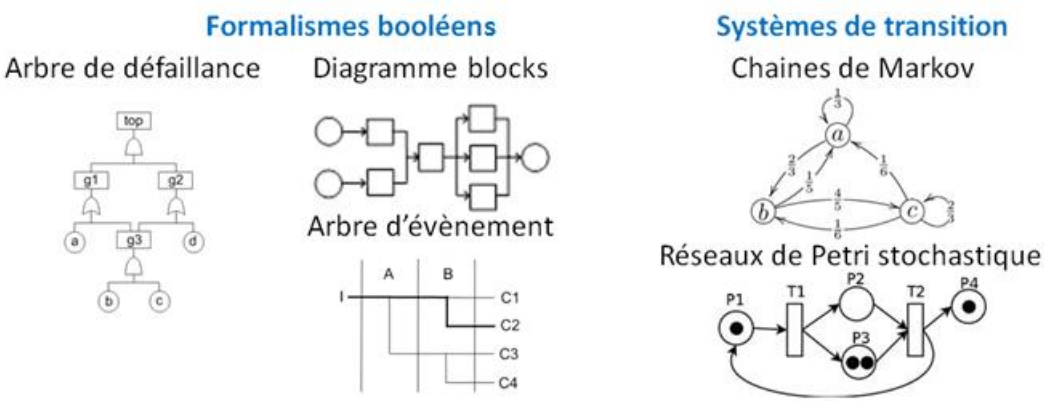


Figure 6 Formalismes classiques employés pour les études de sûreté de fonctionnement [7]

L'équipe AltaRica propose une comparaison [71] des formalismes (cf. Table 4) selon les propriétés attendues des études en sûreté de fonctionnement :

- Modélisation à base d'évènements. Les ingénieurs tentent de capturer des scénarios d'échecs ou d'accidents qui font évoluer l'état du système ;
- Capacité de composition implicite des modèles. Le modèle du système devrait être un assemblage structuré de composants et d'évènements. Les états du système devraient être implicites pour éviter des problèmes d'explosion combinatoire lors des analyses ;
- Capacité de représentation hiérarchique. Les modèles de systèmes devraient être obtenus en composant des modèles de sous-systèmes ou plusieurs vues du système ;
- Capacité de contrôle à distance. Les formalismes devraient être capables d'appliquer des interactions à distance entre les composants ;
- Représentation graphique intuitive. Les formalismes devraient être capables de représenter graphiquement le modèle du système pour permettre des discussions entre les spécialistes ;
- Apparence agréable du code. Les formalismes devraient faciliter la saisie et la lecture des modèles.

Table 4 Comparaison des propriétés de formalismes de sûreté de fonctionnement

	Markov chains	Petri Nets	State Charts	Sequence Algebras	GTS/AltaRica
Event based	●	●	●	●	●
Compositional & Implicit	■	◉	◉	◉	●
Hierarchial	■	■	◉	■	●
Remotely Acting	■	■	■	■	●
Graphical	▲	◉	▲	■	▲
Algorithm Friendly	■	■	■	■	◉

■ not suitable ▲ acceptable ◉ good ● very good

Les représentations classiques, les chaines de Markov et les réseaux de Petri ne sont pas les plus adaptés à la modélisation hiérarchique du système. De plus, les propriétés graphiques

(*Graphical*) et algorithmiques (*Algorithm friendly*) restent très techniques et nécessiteront toujours une connaissance large de la discipline et du système.

Plusieurs équipes de recherche ont proposé des outils et des langages permettant de mettre en œuvre une démarche méthodologique qui intègre les représentations classiques de sûreté de fonctionnement. C'est le *model-based safety analysis ou assessment*, MBSA.

***« Analyse et Evaluation de la sûreté de fonctionnement basée sur les modèles (MBSA)**, est l'application formalisée et outillée de la modélisation, permettant à l'ingénieur de sûreté de fonctionnement de gérer ses modèles et ses activités tout au long de son processus. »*

Le MBSA, par rapport au MBSE, ne s'oppose pas au « document-based ». En effet, les modèles de sûreté de fonctionnement existaient avant l'ère du MBSE. Le MBSA tente de rendre interopérable les études et analyses en proposant des paradigmes de plus haut niveau d'abstraction.

3.5. NORMES

Le référentiel normatif de la sûreté de fonctionnement est imposant. La norme la plus générique est l'ISO 61508 [72], « *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems* ». Elle s'applique aux industries traitant de la sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables relatifs à la sécurité.

La Figure 7 représente une arborescence de quelques normes relatives à la sûreté de fonctionnement. Elle décline les normes par secteur industriel. Les normes du secteur aéronautique sont définies à part. Elles sont définies et maintenues par d'autres organismes, notamment la SAE Aerospace.

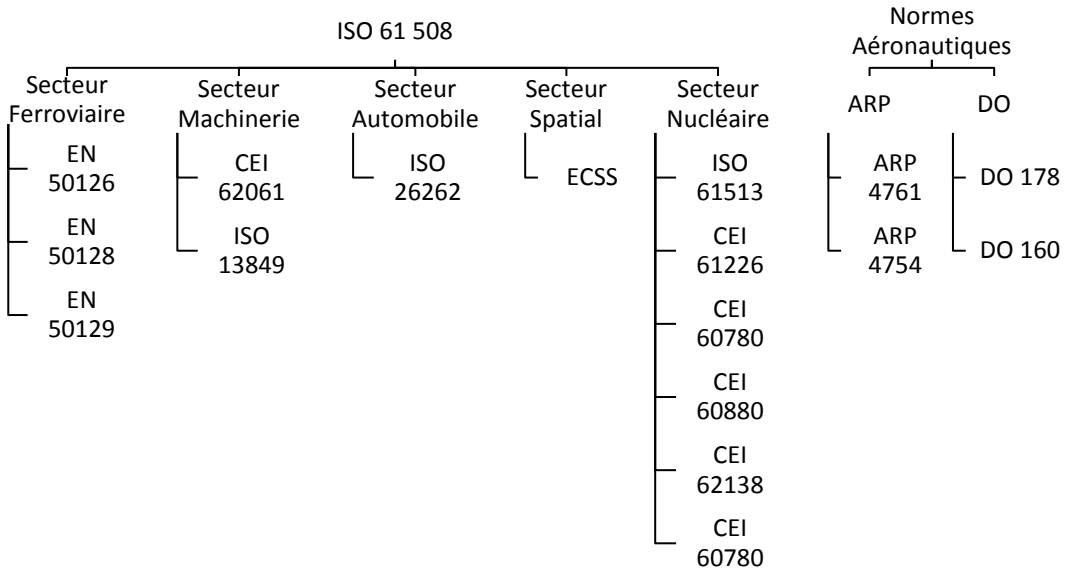


Figure 7 Extrait du référentiel normatif en sûreté de fonctionnement

L'ISO 61508 [72] se décline par secteurs d'activités : ferroviaire, machinerie, automobile, spatial, nucléaire et d'autres.

3.6. LES LOGICIELS

De la même manière que pour les outils de modélisation d'architecture système, nous avons comparé les outils de sûreté de fonctionnement utilisés par les industriels, cf.Table 5. Pour limiter le nombre d'outils cités, nous nous intéresserons seulement aux outils qui déroulent une démarche MBSA. Nous avons cependant ajouté un critère : le(s) secteur(s) industriel(s) dans lequel (lesquels) les logiciels sont les plus employés.

Table 5 Comparaison des logiciels de sûreté de fonctionnement

Nom	Editeur	Environnement	Langage et Formalismes	License	Secteur le plus utilisé
Risk Spectrum [31]	Lloyd´s Register Consulting	Environnement de modélisation et de calcul de sûreté de fonctionnement	Formalisme Booléen et Système de transition	Payante	Energie Nucléaire et Pétrolier
Simfia_V3 [32]	Apsys - Airbus Group	Environnement de modélisation et de simulation basé sur SysML compatible AltaRica Data Flow	**AltaRica Data Flow** : Formalisme Booléen et Système de transition	Payante	Aéronautique, Défense
Grif [33]	Satodev	Environnement de modélisation et de simulation sûreté de fonctionnement	Formalisme Booléen et Système de transition	Payante	Energie pétrolière
Cecilia OCAS [34]	Dassault Aviation	Environnement de modélisation et de simulation basé sur une version AltaRica	**AltaRica (version Dassault) :** Formalisme Booléen et Système de transition	Disponible pour les projets Dassault Aviation	Aéronautique, Défense
Sophia, [73], [11], [74]	CEA	Environnement de modélisation sûreté de fonctionnement basé sur SysML compatible AltaRica 3.0, NuSMV (Logiciel de R&D)	**SysML** + **Profils** Safety Analysis **Sophia** : Formalisme de structuration + génération AltaRica	Indisponible actuellement	Spécialisation disponible (Automobile, Robotique, …)
OpenAltaRica platform [35]	IRT SystemX	Environnement de modélisation et de simulation sûreté de fonctionnement basé sur AltaRica 3.0 (Logiciel de R&D)	**AltaRica 3.0** [70, 75] **:** S2ML [76] et GTS [77]	Gratuit et Code source disponible aux partenaires	Aéronautique, Défense.

Les logiciels de sûreté de fonctionnement se différencient par le langage implémenté et les analyses qu'ils proposent. Pour aller plus loin, il faut s'intéresser plus en détail aux modèles et aux programmes qui exécutent des démarches MBSA.

3.7. LES LANGAGES DE MODELISATION MBSA

Les langages de modélisation dédiés à la sûreté de fonctionnement sont très nombreux. La thèse ne référence que les langages de sûreté de fonctionnement reconnus et implémentés par des outils industriels qui considèrent des représentations à des niveaux d'abstraction système.

[31] [web31] Lloyd's Register group. "Risk Spectrum Tool", sur le site Riskspectrum. Consulté le 31 août 2017. http://www.riskspectrum.com/en/risk/

[32] [web32] Apsys Airbus. "Simfia V3 Tool", sur le site Apsys Airbus. Consulté le 31 août 2017. https://www.apsys-airbus.com/digital-software/#SIMFIA

[33] [web33] Satodev. "Grif Tool ", sur le site Satodev. Consulté le 31 août 2017. http://grif-workshop.fr

[34] [web34] Labri. "Altarica Project, History and dialects", sur le site Labri. Consulté le 31 août 2017. https://altarica.labri.fr/wp/?page_id=23

[35] [web35] IRT System X. "OpenAltaRica Tool ", sur le site IRT System X. Consulté le 31 août 2017. http://openaltarica.fr

FSAP/NuSMV-SA [78]***:*** est un outil d'analyse de sûreté de fonctionnement. FSAP est une plateforme de représentation graphique du système qui fournit une représentation graphique formelle et NuSMV2 est le moteur d'analyse. FSAP/NuSMV-SA utilise des modèles NuSMV pour représenter le système. Ces modèles définissent les modes de défaillance et fournissent des solutions pour :

- La construction des modèles de fautes considérés comme un modèle de connaissance de sûreté. Des bibliothèques par défaut définissent des modes de défaillances génériques comme : *stuck-at, inverted, non_determinism, ramp_down et glitch* ;
- L'injection automatique de pannes ou l'extension de modèles. Une fois les modes de défaillance définis, l'utilisateur peut injecter automatiquement des pannes dans le modèle du système pour créer un nouveau modèle étendu. Ce modèle étendu tient compte du comportement dégradé du système. Il peut ensuite être utilisé pour l'évaluation de la sûreté du système ;
- L'analyse automatisée des arbres de défauts. FSAP génère automatiquement des arbres de défaillance une fois le système et le modèle de défauts spécifiés. Ces arbres permettent dans un second temps d'identifier la liste des coupes minimales.

xSAP [79] ***:*** est un outil d'analyse de sûreté de fonctionnement basé sur des modèles pour les systèmes de transitions synchrones aux états finis et infinis. Il prend en charge l'évaluation probabiliste des arbres de défaillance, des analyses de propagation des défaillances à l'aide des graphes de propagation temporelle et de l'analyse des causes communes. xSAP a été utilisé dans plusieurs projets industriels et dans un projet de R&D impliquant FBK et la société Boeing.

Figaro [80]***:*** est un langage développé en 1990, implémenté dans l'outil KB3 et FIGSEQ par les équipes d'EDF R&D. Ce langage à deux niveaux permet d'abord la construction d'une base de connaissance puis du modèle d'architecture. Les analyses exécutables sur un modèle FIGARO sont : la génération d'arbres de défaillance, la génération de séquences, le calcul de coupes minimales, le calcul de la probabilité des évènements redoutés et l'identification de facteurs d'importance.

Le langage Figaro est open-source cependant les outils implémentés sont propriété d'EDF R&D et sont exclusivement utilisés par les industries de l'énergie et du gaz (IEG).

AltaRica 3.0 [70]**,** [71] ***:*** est un langage de modélisation de haut niveau dédié à l'analyse de la sûreté de fonctionnement. Il succède aux versions AltaRica et AltaRica dataflow. Les formalismes «classiques» tels que les arbres de défaillance, les chaînes de Markov et les réseaux de Petri sont trop éloignés des représentations du système étudié. AltaRica répond à ce problème et s'appuie sur un paradigme mathématique puissant et très expressif : système de transition gardée (Gard Transition System, GTS). Celui-ci permet une construction formelle de modèles dysfonctionnels au niveau du système. De plus, l'Association AltaRica (association savante autour d'AltaRica 3.0) fournit divers outils utilisant AltaRica 3.0 ou des modèles dérivés (au format GTS, OpenPSA, …). Les outils d'évaluation sur AltaRica sont : un compilateur d'arbres de défaillance, un générateur de séquences, un générateur de chaînes de Markov, un simulateur stochastique, un model-checker, un simulateur pas-à-pas etc. Ce langage AltaRica 3.0 et les outils associés sont la base du projet OpenAltaRica dont l'objectif est de fournir une plateforme logicielle intégrée dédiée à l'analyse de la sûreté de fonctionnement de systèmes complexes.

En complément des langages présentés ci-dessus, nous présentons aussi des outils de calculs d'analyses de sûreté de fonctionnement. Ceux-ci ne sont pas véritablement des outils MBSA mais ils s'alignent et sont conformes à cette approche.

HIP-HOPS (Hierarchically Performed Hazard Origin and Propagation Studies) [81] : est une méthode outillée d'analyse de sûreté de fonctionnement issue de techniques classiques tout comme les outils précédemment cités.

La méthode permet une évaluation intégrée d'un système complexe du niveau fonctionnel en tenant compte des modes de défaillance des composants. La chaine d'outils HIP-HOPS commence très en amont dans le cycle de développement. Elle permet de construire un modèle comportemental dysfonctionnel à partir du tableau AMDEC (orientée matériel et logiciel) et de générer des d'arbres de défaillance. HIP-HOPS est implémenté dans l'outil Safety Argument Manager (SAM).

Galileo : est un outil de modélisation d'arbre de défaillance qui intègre une méthodologie d'analyse, *DIFTree*. Il construit des arbres de défaillance dynamiques pour permettre la modélisation de systèmes tenant compte de la tolérance aux pannes. La méthodologie d'analyse DIFTree (Dynamic Innovative Fault Tree) [82] combine des techniques statiques et dynamiques d'analyse des arbres de défauts en utilisant une approche modulaire.

Galileo ne permet pas la génération d'arbres de défaillance. Cependant, il peut s'interfacer avec un outil automatisé de génération d'arbres de défaillance, par défaut FSAP. Galileo effectue des analyses quantitatives en caractérisant la distribution des évènements défaillants.

4. INTERACTIONS MULTIDISCIPLINAIRES

L'architecte système et l'ingénieur de sûreté de fonctionnement ont besoin de concilier des vues de leurs modèles. Cependant, celles-ci sont hétérogènes. Il est donc nécessaire de proposer des solutions pour assurer la cohérence globale, le maintien de la traçabilité et la création de vues multidisciplinaires dans ce contexte. Plusieurs équipes de recherche [83] apportent des réponses aux problèmes d'interactions entre l'architecture système et la sûreté de fonctionnement. Elles peuvent être regroupées en trois approches principales :

- L'approche « d'analyses » conjointes d'architecture système et de sûreté de fonctionnement ;
- L'approche orientée d'une discipline vers l'autre ;
- L'approche collaborative multidisciplinaire.

Les approches d'analyses conjointes entre l'architecture système et la sûreté de fonctionnement ne seront pas présentées dans ce mémoire. On peut toutefois renvoyer aux travaux de thèse de Pierre Mauborgne [84] chez PSA. Dans sa thèse, il propose d'enrichir les activités de l'architecte système pour prendre en compte les aspects dysfonctionnels qualitatifs lors de la définition des exigences système et de la conception des architectures logiques. Il précise que les aspects dysfonctionnels quantitatifs, essentiels pour assurer la vérification et validation des exigences « Safety », seraient toujours réalisés par les spécialistes « Safety ». Il a présenté un modèle conceptuel, pour chaque vue d'ingénierie [85], [86], permettant de préciser les relations entre concepts clés et de lever les ambiguïtés sur les termes utilisés par ces deux domaines, ainsi que des processus enrichis. Par contre, il ne propose pas de modèles ou de mécanismes permettant d'assurer les interactions entre les architectes système et spécialistes « Safety ».

4.1. APPROCHE ORIENTEES

L'approche « orientée » provient essentiellement d'équipes de recherche des communautés OMG et SAE International. Elle est basée sur les techniques d'annotations de modèles mais elle ne s'appuie pas toujours sur le même langage. Elle considère un modèle d'architecture tel quel puis ajoute des propriétés dysfonctionnelles aux composants des modèles. Les modèles résultants sont utilisés pour mener des études de sûreté de fonctionnement.

L'approche impose à l'ingénieur de sûreté de fonctionnement d'adopter le point de vue et le niveau d'abstraction de l'architecte système pour mener ses analyses. Les principaux travaux appliquant cette approche sont présentés ci-dessous :

- **SysML + Profil :**

  - **Sophia** [74]**,** [73]**:** Sophia est un outil intégré à l'environnement de modélisation Papyrus. Il propose une méthodologie MBSA (« gros grains ») en s'appuyant sur SysML comme modèle d'architecture du système. Il offre plusieurs profils (technique d'annotation UML) selon les besoins d'études (SHA, FMEA, FTA, MCs) à saisir par l'ingénieur.

    Lorsque les études deviennent complexes ou du moins qu'elles ont besoin d'une puissance de calcul importante, Sophia exécute des transformations de modèles vers des langages formels tel que AltaRica 3.0 [70]**,** [71] et NuSMV.

  - **MeDISIS** [87]**,** [88]**:** La méthode MeDISIS tire pleinement partie de cette approche et tente de fournir des outils pour faciliter le stade de la spécification et l'intégration de la sûreté de fonctionnement au début du processus de conception. MeDISIS s'appuie sur le langage SysML pour mener une démarche MBSE en sûreté de fonctionnement. Grace à l'utilisation des profils, la méthode outillée transforme le modèle d'architecture système vers des formalismes comme AltaRica Data flow (version antérieure à AltaRica 3.0) et AADL.

- **EAST-ADL + Error annex** [89], [90] **:**
  - Un cadre de modélisation MBSE, développé lors des projets ATTEST puis MAENAD, est basé sur EAST-ADL. Le cadre complète le langage EAST-ADL par une méthodologie d'analyse de risques conforme à la norme ISO26262 en utilisant le concept d'annotation « Error Annex » d'EAST-ADL. Il permet de mener des analyses de sûreté de fonctionnement spécifiques au secteur automobile.

- **AADL + Error annex :**
  - Marco Bozzano and al. [91] et Julien Brunel and al. [92] proposent des cadres de modélisation pour l'analyse et l'évaluation de la sûreté de fonctionnement. Ceux-ci tiennent compte des aspects temps réel et sont spécifiques aux systèmes aérospatiaux. Ils permettent de construire un modèle d'architecture AADL [5] et d'incorporer un modèle « Erreur Annex » (ou « Error Annex 2 »). Ces modèles sont transformés vers le langage AltaRica 3.0 [70] pour mener des analyses de sûreté de fonctionnement. Ces travaux ont donné lieu à des applications dans le secteur aérospatial, projet financé par l'Agence spatiale européenne [92].

- **SAML** [93] **:**
    - C'est un langage de modélisation permettant de mener des analyses de sûreté de fonctionnement. Il s'interface avec le langage SysML. Les méthodes implémentées peuvent répondre aux questions logiques et probabilistes. L'outil permet de créer intuitivement des exigences de sûreté (objectif de sûreté), non déterministes et déterministes pour mener des analyses formelles. Le modèle est ensuite transformé automatiquement dans le langage d'entrée d'un moteur de vérification des propriétés.

4.2. APPROCHE COLLABORATIVES MBSE ET MBSA

L'approche « collaborative multidisciplinaire » regroupe les travaux tentant de concilier les approches MBSE et MBSA. Elle est collaborative car elle accompagne un travail de groupe entre différentes disciplines et entre différents individus, pour l'élaboration d'un résultat commun. L'approche ne se préoccupe pas de l'orientation des interactions, elle rend tous les scénarios possibles. Elle n'impose pas aux disciplines d'ingénierie de point de vue ou de niveau d'abstraction. C'est la différence majeure avec les approches précédentes. L'approche met en cohérence des modèles hétérogènes en se focalisant sur leur contenu. Elle permet de raisonner directement sur les éléments et leurs sémantiques. Les principaux travaux respectant cette approche sont présentés ci-dessous :

Fédération de modèles [94] **:** est une technique permettant d'assurer des cohérences, le maintien de la traçabilité et la création de vues intersectorielles dans un contexte industriel. L'approche fédération de modèles dans l'outil OpenFLEXO appuie la construction de vues conceptuelles sur les modèles et les outils existants. Elle a été appliquée à : la composition du modèle dans les domaines experts, l'alignement hétérogène (méta) et l'alignement des modèles.

Conception Collaborative [95] **:** Le développement des produits dans les entreprises multinationales implique de nombreux experts de multiples domaines de langues différentes et ce tout au long du cycle de vie des projets. Laurent Wouters propose une technique qui appuie les activités de conception collaborative en comblant les lacunes entre les différents domaines d'expertises ainsi que les langues.

Synchronisation de modèles : Avant cette thèse, aucune approche ne proposait d'utiliser les techniques de synchronisation de modèles pour conduire des interactions multidisciplinaires. Cependant, des approches dans d'autres contextes, d'autres objectifs et d'autres modèles exécutent des techniques similaires.

Dans le domaine de l'informatique et des télécoms, la réconciliation de données divergentes est un des problèmes clefs. Gérald Oster and al. [96] proposent d'utiliser des transformées opérationnelles pour raisonner sur la synchronisation de données divergentes. Ils ont défini un algorithme et des fonctions de transformation qui exécutent la réconciliation d'un système de fichiers.

Une équipe allemande de l'université Carl von Ossietzky d'Oldenbourg [97] propose des techniques pour améliorer les interactions entre les modèles opérationnels et fonctionnels. Elle propose de synchroniser des modèles de manière bidirectionnelle. Cependant, les techniques sont assez éloignées de notre approche.

Le projet MOISE [92] de l'IRT Saint Exupéry de Toulouse propose un processus de synchronisation entre l'architecture système et la sûreté de fonctionnement. Il met en œuvre

leur processus entre deux équipes expérimentales en s'appuyant sur un cas d'étude aéronautique.

4.3. TRAVAUX CONNEXES

Cette partie présente des travaux de recherches additionnelles. D'une certaine manière, elles contribuent à traiter la problématique. Elles utilisent des techniques très variées issues de réflexions connexes.

4.3.1. LES ONTOLOGIES

Il existe de très nombreuses approches pour aligner des ontologies résultant de correspondances sémantiques.

Ontologie [36]*: en informatique, une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts.*

Patrick Arnold [98] propose une approche qui supporte plusieurs relations de correspondance telles que l'égalité, la relation ontologique et les relations de relations ontologiques. La stratégie d'enrichissement utilise plusieurs approches linguistiques et des heuristiques pour identifier les correspondances.

Dans une méthodologie MBSE, l'étape de conception d'un produit implique la création de modèles dans plusieurs domaines. Mathieu Perin et Laurent Wouters [99] proposent de résoudre l'intégration de modèles hétérogènes grâce à l'utilisation d'ontologies OWL (Web Onthology Language). Ils tiennent compte de la sémantique comportementale des modèles spécifiques aux domaines et fournissent une solution basée sur le langage xOWL, utilisée pour l'expression des comportements au sein des ontologies.

L'utilisation de différents langages dans le développement de logiciels rend l'intégration de la solution difficile. Kappel et al. [100], proposent un processus qui fait évoluer semi automatiquement les métamodèles en ontologies. Cela se fait en ajoutant implicitement des concepts d'ontologie et permet d'établir un unique niveau conceptuel pour construire de meilleurs mappings. Ces derniers peuvent à leur tour servir de base pour la transformation de modèles hétérogènes.

4.3.2. REGLES DE COHERENCE AVEC UML

Les diagrammes UML [26] décrivent différentes vues d'un logiciel. Ces diagrammes sont fortement dépendants et doivent donc être cohérents les uns avec les autres. Il est donc primordial que les règles de cohérence soient définies et que les incohérences soient détectées, analysées et corrigées. Les travaux de Damiano Torre [101] fournissent un ensemble de règles de cohérence des modèles UML.

[36] [web36], Entretien avec Bruno Bachimont (Juillet 2006). "Qu'est-ce qu'une ontologie?", sur le site Technolangue. Consulté le 31 août 2017. http://www.technolangue.net/imprimer.php3?id_article=280

4.3.3. TISSAGE DE MODELES

La transformation du modèle est une opération centrale de l'IDM. Cependant, il est souvent nécessaire de créer des liens entre les modèles (mapping). Ces liens peuvent être capitalisés par un modèle de tissage (Model Weaving). Didonet Del and al. [102] proposent un prototype de tissage de modèles dans le cadre de la construction d'AMMA, une plate-forme de modélisation IDM.

4.4. POSITIONNEMENT DES TRAVAUX DE THESE

Les modèles employés durant le cycle de conception d'un système complexe sont très nombreux et mettent en avant de multiples préoccupations. De plus, exploités par de nombreuses disciplines, ces modèles sont hétérogènes. Cependant, leurs contenus sont fortement dépendants (les uns des autres) puisqu'ils traitent du même système étudié ISO 42010 [8]). Les interactions entres les disciplines amènent à solliciter certaines disciplines plus tôt dans les processus.

Pour être capable d'établir une cohérence entre les modèles, il est nécessaire de les abstraire à un niveau d'abstraction commun et de comparer ces abstractions. Ainsi, il est possible d'identifier les écarts entre les éléments issus de deux modèles. A noter qu'une incohérence détectée par la synchronisation constitue déjà un résultat significatif car elle met en évidence le fait que les ingénieurs aient identifié précisément des écarts sémantiques entre leurs modèles.

La thèse est axée sur la formalisation des concepts puis la proposition d'une méthodologie de synchronisation de modèles. Elle propose plusieurs manières de mettre en œuvre et de configurer la synchronisation de modèles au sein d'un projet. La méthodologie permet d'échanger des propositions sur l'architecture du système en utilisant les principes d'abstraction et de comparaison des modèles. Elle permet ainsi de construire des modèles de cohérence tout au long des processus.

Les travaux de cette thèse répondent aux problématiques d'interactions multidisciplinaires entre des disciplines d'ingénierie ayant des démarches Model-Based, notamment : en conception d'architecture système (MBSE) et en analyse de la sûreté de fonctionnement (MBSA).

Elle propose une approche collaborative et itérative à plusieurs niveaux conceptuels entre les modèles d'architecture système et les modèles de sûreté de fonctionnement. Elle guide les ingénieurs au travers d'un dialogue pour l'élaboration de cohérences entre les modèles employés tout le long du cycle de développement d'un système complexe. L'approche fait appel à des activités de synchronisation incrémentale qui permettent progressivement de résoudre les incohérences identifiées par la recherche de compromis qui feront évoluer les modèles respectifs des disciplines.

Chapitre II CADRE CONCEPTUEL DE SYNCHRONISATION DE MODELES

Ce chapitre définit le cadre conceptuel nécessaire à l'application de la synchronisation de modèles pour mener des interactions multidisciplinaires. Le cadre conceptuel repose sur l'état de l'art, sur des standards et sur des échanges avec des experts de sociétés comme : DGA, SAFRAN, SAFRAN Landing System, Airbus, PSA, DCNS et Thales.

Dans un premier temps, le chapitre présente le périmètre du cadre conceptuel alimenté par une analyse opérationnelle du processus de synchronisation de modèles. Il détaille ensuite le plan du chapitre. Dans un second temps, il présente en détaille les concepts et formalismes utiles aux étapes de synchronisation de modèles. Les concepts et les travaux de formalisation proposés pourront être repris par de futurs travaux de recherche et éventuellement par de futures applications industrielles.

1. PERIMETRE DU CADRE CONCEPTUEL

Le cadre proposée est celui d'un processus de conception et de réalisation d'interactions multidisciplinaires. Il englobe plusieurs sous-processus, représentés dans la Figure 8 :

- Les processus des disciplines d'ingénierie considérées. Ils seront suivies par les disciplines d'ingénierie ;
- un processus d'intégration des autres processus. C'est le processus de synchronisation de modèles, il est composé de plusieurs étapes (ou tâches).

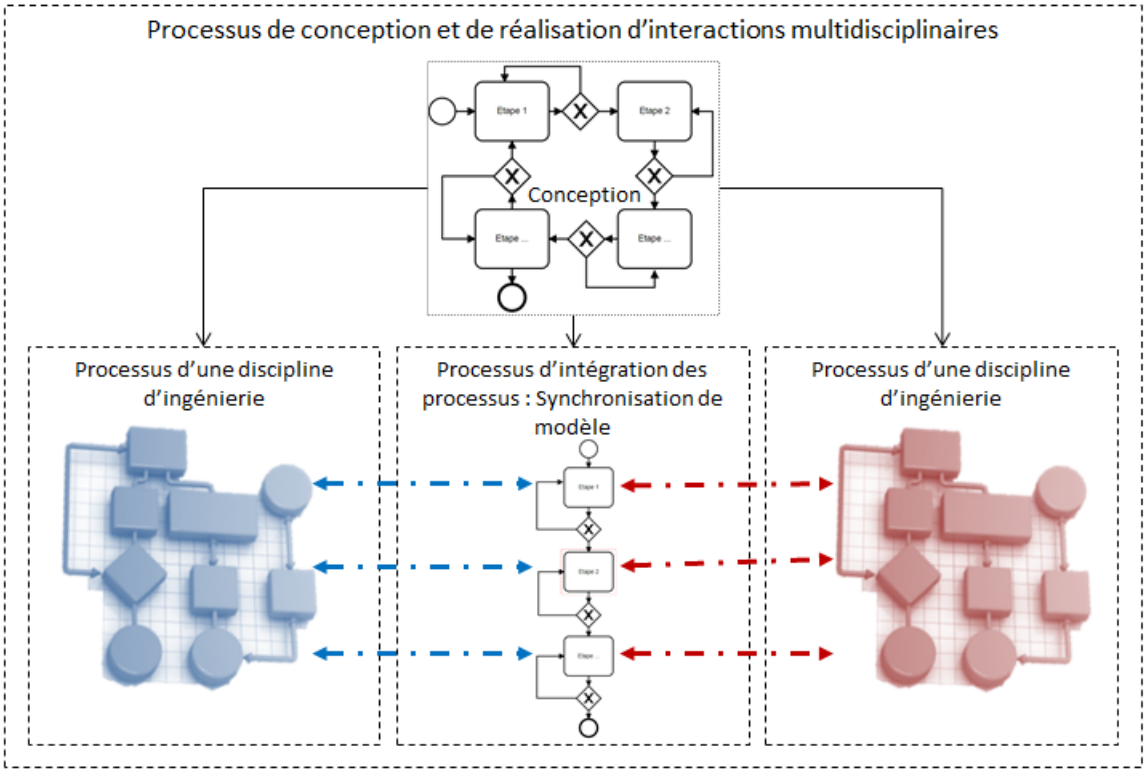


Figure 8 Périmètre du cadre conceptuel

L'analyse opérationnelle qui suit est effectuée dans le but de définir les fonctions et les étapes de conception d'un processus de synchronisation de modèles. Elle définit successivement le sujet de l'analyse, les parties prenantes, les étapes de développement et les relations des parties prenantes avec le sujet.

1.1. PARTIES PRENANTES DU PROCESSUS DE CONCEPTION

Le cadre de l'analyse est un « processus de conception et de réalisation d'interactions multidisciplinaires ». Ce processus configure, exécute et capitalise un ensemble d'interactions multidisciplinaires. Le cadre est conforme à la théorie de la synchronisation de modèles (cf. 1. Objectifs des travaux). Les parties prenantes sont : le sponsor, le responsable de synchronisation, les disciplines d'ingénierie impliquées dans les interactions multidisciplinaires.

Le terme sponsor est repris du référentiel TOGAF 9.1. Cependant, ce référentiel utilise le terme sans le définir. Nous proposons donc la suivante :

Sponsor : *Individu ou équipe décisionnelle de l'entreprise qui formule une demande pour mettre en place une démarche collaborative entre des disciplines d'ingénierie. Cette entité décisionnelle doit financer les activités liées à sa demande.*

Responsable de synchronisation : *Individu ou équipe en charge de mettre en place une solution concrète de synchronisation de modèles au sein de sa structure. Il joue le rôle de l'architecte système (cf. Chapitre I2.2. Missions de l'architecture système), cependant il se positionne au niveau de l'architecte d'entreprise [49], [103].*

Discipline d'ingénierie *: C'est une partie prenante active. Elle est chargée de répondre à un besoin d'étude d'un système sous certaines préoccupations.*

Voici quelques exemples de disciplines d'ingénierie : l'ingénierie des exigences, l'architecture système, la sûreté de fonctionnement, le soutien logistique intégré, la conception mécanique, la conception électronique, le développement logiciel, etc.

1.2. ETAPES DU PROCESSUS DE SYNCHRONISATION

Les parties prenantes ont toutes des interactions avec le processus de synchronisation. Pour caractériser ces interactions, le cycle de développement est défini en quatre étapes successives :

- **Etape de lancement**, elle permet de formuler une demande de collaboration entre des disciplines d'ingénierie.
- **Etape de conception,** elle permet de configurer les interactions que les disciplines souhaitent conduire entre elles pendant les projets.
- **Etape de réalisation,** elle permet de conduire des interactions définies dans l'étape précédente. Elle s'applique tout au long du projet de conception.
- **Etape de post-synchronisation,** elle permet d'exploiter et d'évaluer des traces issues des interactions réalisées.

Les activités envisagées et les rôles des parties prenantes, pour chaque étape, sont définies ci-dessous (de manière informelle) :

A l'étape de lancement, le sponsor initie une demande motivée d'action collaborative qu'il finance. Le responsable de synchronisation spécifie ensuite la demande et caractérise le besoin du sponsor.

A l'étape de conception, le responsable de synchronisation définit l'environnement et les interactions qui seront conduites, en proposent des solution(s) à plusieurs niveaux d'abstraction. Il dialogue avec les disciplines d'ingénierie pour comprendre leurs activités, leurs pratiques, les modèles utilisés et les objectifs de leurs études. Le sponsor suit les propositions du responsable de synchronisation et évalue la ou les solution(s) selon le contexte financier et décisionnel.

A l'étape de réalisation, les disciplines d'ingénierie réalisent les interactions définies durant l'étape de conception. Pour chaque interaction, les disciplines discutent entre elle en s'appuyant sur les représentations produites ou utilisées du système. Elles tentent de construire ensemble des correspondances entre le contenu commun de leurs modèles. Le responsable de synchronisation peut aider et accompagner les disciplines dans ces activités.

A l'étape de post-synchronisation, les disciplines d'ingénierie peuvent réutiliser les résultats de comparaison et la traçabilité des interactions pour alimenter les documentations et les livrables (tel que le dossier de sécurité) du projet de conception. Le responsable de synchronisation étudie également les résultats des interactions pour fournir un retour qualitatif et quantitatif au sponsor. Il peut également transmettre une demande d'évolution pour raffiner, modifier ou ajouter des interactions pour de nouvelles applications dans de futurs projets. Le sponsor choisit de financer cette évolution ou non.

Les interactions avec les parties prenantes ont été représentées sous la forme d'un diagramme de cas d'utilisation [17] du « processus de conception d'interactions multidisciplinaires », Figure 9.

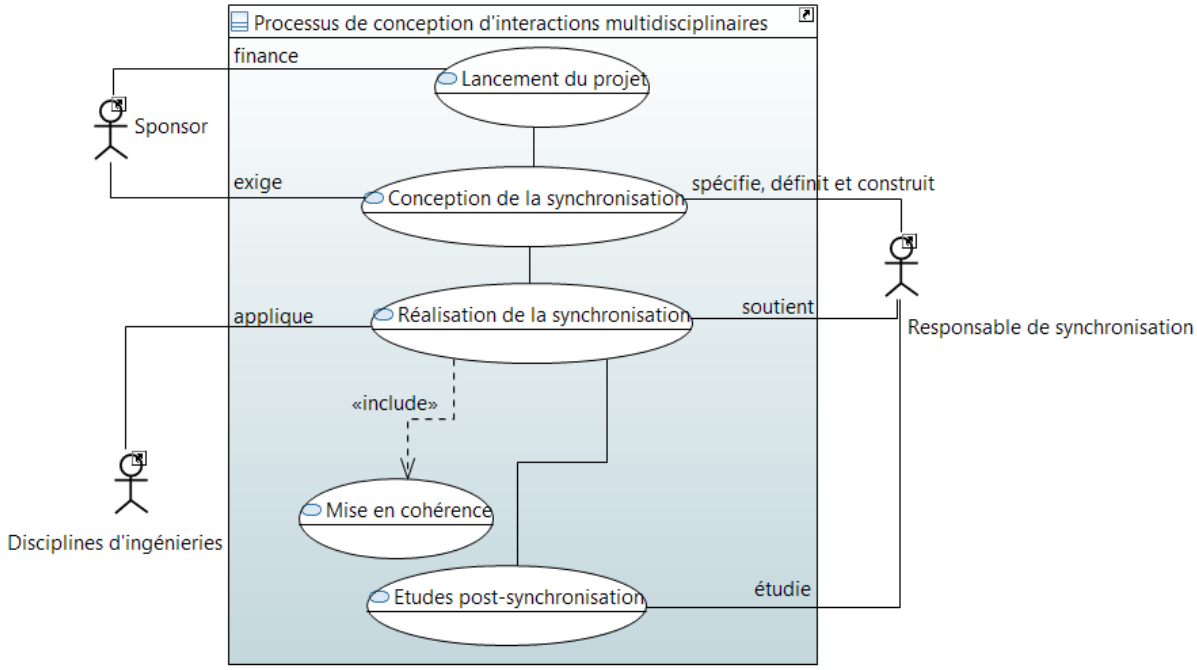


Figure 9 Diagramme de cas d'utilisation du processus de synchronisation de modèles

1.3. PROCESSUS DE CONCEPTION ET FONCTIONS DE SYNCHRONISATION

A partir de l'analyse opérationnelle et de la définition de la synchronisation de modèles rappelée ci-dessous, les fonctions nécessaires à sa mise en œuvre ont été définies.

*La **synchronisation de modèles** est un cadre théorique permettant d'établir une correspondance entre les contenus de modèles. Elle part du constat que les modèles ne peuvent pas être comparés directement dans leur formalisme respectif du fait de leur hétérogénéité. Elle propose donc d'abstraire le contenu des modèles dans un formalisme commun et de comparer ces abstractions. L'objectif de cette comparaison est d'assurer la cohérence des modèles, ou tout au moins, de permettre aux différentes disciplines d'ingénierie d'identifier les incohérences présentes entre leurs modèles.*

Ce cadre théorique permet de construire et de maintenir des modèles hétérogènes représentants le même système de manière cohérente.

Les fonctions principales sont les suivantes :

- Définir et configurer la synchronisation :
  - Capitaliser les besoins d'interactions multidisciplinaires ;
  - Définir et configurer les interactions multidisciplinaires ;
  - Ordonnancer les interactions multidisciplinaires ;
- Appliquer la synchronisation :
  - Réaliser les abstractions des modèles ;
  - Accompagner la comparaison des modèles ;
  - Réaliser des concrétisations vers les modèles d'origine ;

Les fonctions secondaires sont :

- Définir l'environnement et les parties prenantes ;
- Définir et manipuler les contenus des modèles qui seront pris en compte par les applications de synchronisation ;
- Construire et manipuler la traçabilité des évolutions des modèles liées aux interactions effectuées.

Les fonctions supports sont :

- Définir une démarche de mise en œuvre en entreprise ;
- Définir les principes de synchronisation.

Plusieurs étapes ont été définies, elles permettent de satisfaire les fonctions identifiées. Pour chaque étape, des concepts, des métamodèles et des techniques sont formalisés. La Figure 10 présente le plan du chapitre et l'organisation des étapes.

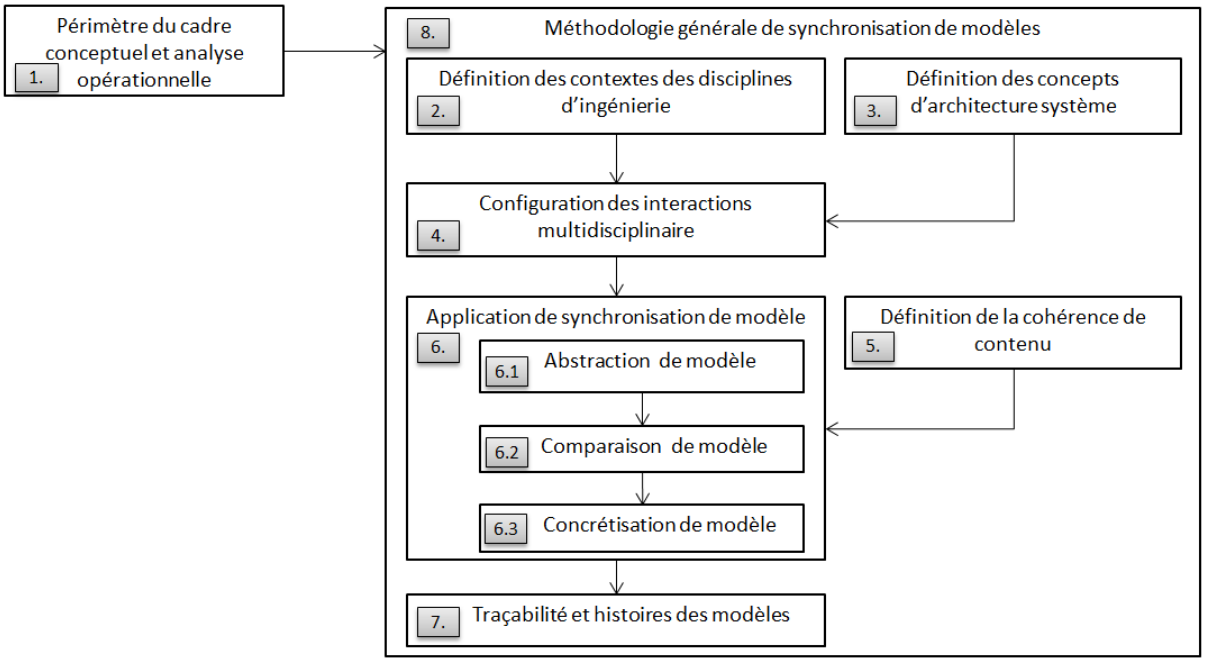


Figure 10 Plan du chapitre II et étapes du processus de synchronisation de modèles

2. DEFINITION DES DISCIPLINES D'INGENIERIE

Pour envisager des interactions entre disciplines, il est nécessaire de caractériser les modèles utilisés par chaque discipline d'ingénierie. La définition des contextes répond à ce besoin. Elle tente de lier les processus, les méthodes et les représentations structurelles du système.

Plusieurs concepts sont repris de standards (ISO 42010 [8] et ISO 15288 [35]) ou sont définis à l'occasion. Un métamodèle et une représentation graphique pour la visualisation des contextes sont proposés.

La définition des contextes permet ensuite d'identifier les instants les plus adéquats, dans les processus, pour effectuer des interactions.

Les concepts nécessaires à la définition des contextes sont définis en deux parties. La section présente, dans un premier temps, les concepts liés aux notions d'architecture et de modèle. Puis, dans un second temps, elle présente les concepts liés aux disciplines d'ingénierie.

2.1. CONCEPTS D'ARCHITECTURE ET DE MODELE

L'ISO 42010 [8] propose une standardisation des concepts d'architecture. Elle tente d'harmoniser des termes utilisés dans d'autres normes. Elle propose (Figure 11) un modèle reliant la description d'architecture aux concepts de systèmes et de parties prenantes.

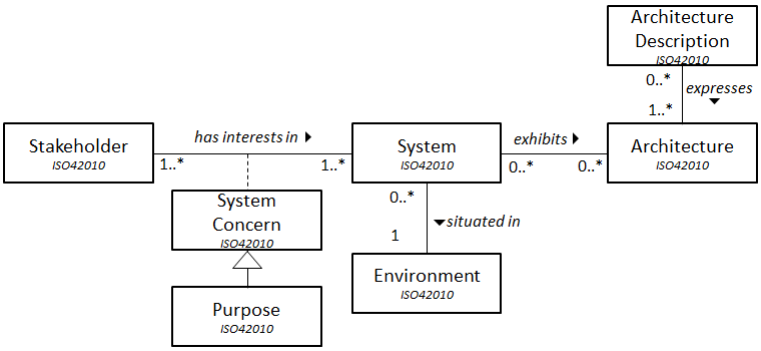


Figure 11 Contexte du concept architecture description ISO 42010

Le concept de « Partie prenante » est définis Chapitre I1.2

***Système** : ISO 15288 [35] : combinaison d'éléments interagissant entre eux qui sont organisés pour atteindre un ou plusieurs objectifs.*

Note 1 : Un système est parfois considéré comme un produit ou comme les services qu'il fournit.

Note 2 : Un système complet comprend tous les équipements, les installations, le matériel, les programmes informatiques, le microprogramme, la documentation technique, les services et les personnels requis pour les opérations et le support, dans la mesure nécessaire à une utilisation autonome dans son environnement prévu.

***Architecture**, ISO 42010 [8] et TOGAF 9.1 [103] : c'est un concept fondamental qui représente l'ensemble des propriétés d'un système dans son environnement, incarné par ses éléments, ses relations et ses principes de conception et d'évolution.*

***Description d'architecture**, ISO 42010 [8] : c'est le résultat utilisé pour exprimer une architecture.*

L'ISO 42010 [8] distingue l'architecture d'un système de sa description. La description d'architecture est un produit issu d'un travail tandis qu'une architecture n'en est qu'une synthèse. De plus, la norme définit des concepts et un métamodèle (Figure 12) fait le lien entre l'architecture, les points de vue et les modèles.

***Système étudié,** ISO 15288 [35] : le système dont le cycle de vie est à l'étude dans notre contexte.*

***Préoccupation**, ISO 42010 [8] et TOGAF 9.1 [103] : intérêt principal qui est d'une importance cruciale pour les parties prenantes d'un système et déterminante pour l'acceptabilité du système. Une préoccupation peut concerner tous les aspects du fonctionnement, du développement ou de l'exploitation du système, y compris des considérations telles que la performance, la fiabilité, la sécurité, la distribution et l'évolutivité.*

***Point de vue d'Architecture (Viewpoint)** ISO 42010 [8] : c'est un résultat définissant les modalités de construction, d'interprétation et d'utilisation d'une vue d'architecture pour capturer des préoccupations particulières du système.*

***Vue d'architecture (View) :** résultat exprimant l'architecture d'un système sous l'angle des préoccupations spécifiques du système ISO 42010 [8]. Une vue d'architecture est une représentation qui met en évidence une préoccupation (ou une partie d'une préoccupation) identifiée par les parties prenantes sur le système concerné.*

Le point de vue d'architecture peut être considéré comme la définition d'une vue d'architecture du système.

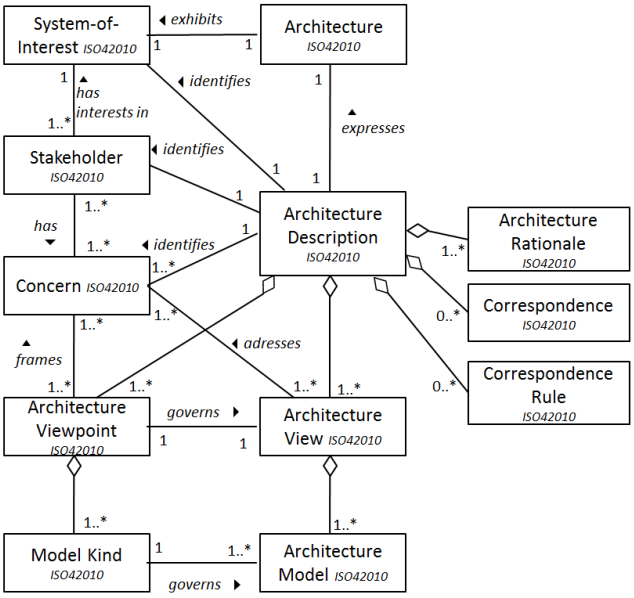


Figure 12 Modèle conceptuel d'une description d'architecture ISO 42010

Une description d'architecture comprend une ou plusieurs vues d'architecture. Une vue d'architecture (ou simplement vue) répond à une ou plusieurs préoccupations des acteurs du système. Une vue d'architecture représente l'architecture du système selon un point de vue d'architecture. Il y a deux aspects à considérer : les préoccupations pour les acteurs et les conventions établies sur les points de vue. Une vue d'architecture est composée d'un ou plusieurs modèles d'architecture. Ce (ces) dernier(s) utilise(nt) les conventions de modélisation appropriées selon les préoccupations.

2.2. CONCEPTS D'UN CONTEXTE DES DISCIPLINES D'INGENIERIE

En complément des concepts précédents, plusieurs autres sont définis. Ils permettent de caractériser un contexte disciplinaire.

***Discipline d'ingénierie** : c'est une partie prenante. Elle possède des préoccupations spécifiques à un domaine d'étude ou d'analyse. Elle intervient sur un système durant son cycle de vie pour répondre à des problématiques non triviales.*

***Contexte d'une discipline d'ingénierie :** c'est une notion abstraite qui inclut toutes les compétences: savoir, savoir-faire et savoir-être spécifiques à une discipline d'ingénierie. Il a un but (ou un ensemble d'objectifs), généralement pour rendre un service, un produit ou un résultat attendu. Le contexte est rattaché à une discipline d'ingénierie.*

***Processus disciplinaire:** il s'agit d'un ensemble d'activités structurées (parfois appelées tâches), qui produisent un service dans un contexte spécifique. Les processus métier sont souvent représentés par l'utilisation de flowchart contenant des séquences d'activités entrelacées par des points de décision et un fork-join. Une séquence d'activités est un enchaînement d'activités orienté selon les transitions.*

Flowchart est un cadre mathématique qui permet de représenter le déroulement d'un processus. Il est très largement utilisé dans de nombreux domaines pour étudier, planifier et communiquer

sur des processus. Il s'appuie sur des représentations graphiques telles que : BPMN [27], Process Flow Diagram (PFD) [104] ou Software Process Engineering Metamodel SPEM [38]. Certains sont liés à d'autres diagrammes, tels que les diagrammes de flux de données DFD [105] ou les diagrammes d'activités d'UML [26].

Les points de décision et les fork-join permettent de représenter la séparation ou le regroupement de séquences d'activités pour caractériser des séquences alternatives ou parallèles (temporellement ou logiquement).

Activité : *c'est l'application d'une méthode particulière à un instant donné durant un processus ou une méthodologie. Selon ISO 9000 (2005) [106] et ISO 15288 [35], l'activité définit clairement les éléments d'entrée et de sortie dont la valeur ajoutée est mesurable. Il peut exister des éléments intermédiaires pour représenter des résultats non-finaux. Ainsi, une activité est définie par un objectif de résultat attendu (qui peut être un sous-objectif d'un processus). Il est possible de considérer une condition (une garde) sur la faisabilité de l'activité en fonction de la maturité des éléments d'entrée et/ou de l'accomplissement des activités en amont.*

Méthode : *c'est un ensemble ordonné de manière logique, de principes, de règles, d'étapes qui constituent un moyen pour parvenir à un résultat attendu. Une méthode emploie des éléments d'entrée et produit des éléments de sortie. Ces derniers sont représentés dans des vues graphiques, textuelles, …*

Remarque : Une méthode répond à un objectif mais elle n'est pas spécifique à une activité ni un contexte métier.

Besoin de synchronisation *: c'est une relation entre plusieurs points de vue d'architecture issus de différents contextes. Il est aussi caractérisé comme un instant identifié entre les processus des disciplines d'ingénierie où la mise en cohérence d'informations est possible. Ce besoin répond à un besoin formalisé de partage de points de vue par des disciplines manipulant des objets ayant des dépendances entre elles.*

2.3. MODELISATION DES CONCEPTS

Cette section organise les concepts permettant de :

- caractériser une activité ;
- définir et représenter le contexte d'une discipline d'ingénierie ;
- construire, caractériser et représenter des besoins de synchronisation.

Pour être homogène avec les standards, tous les métamodèles présentés sont en anglais. A l'inverse, pour mieux illustrer nos propos, les modèles, qui en découlent, sont en français.

2.3.1. METAMODELE D'UNE ACTIVITE

Une activité d'un processus est liée à de nombreux concepts : données d'entrée, de sortie, objectif, méthode, condition etc. Pour clarifier ces derniers, la Figure 13 représente le métamodèle d'une activité :

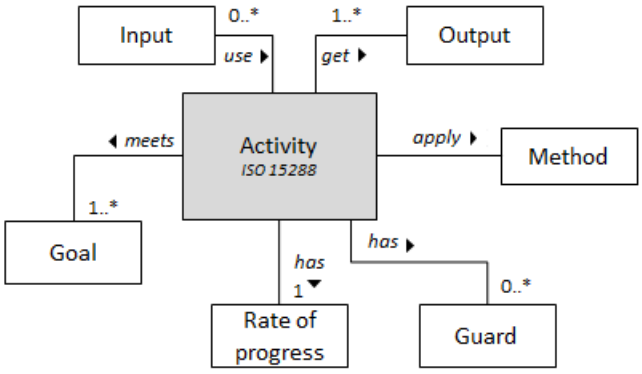


Figure 13 Modèle d'une activité

2.3.2. METAMODELE D'UN CONTEXTE D'UNE DISCIPLINE D'INGENIERIE

La Figure 14 illustre le métamodèle d'un contexte d'une discipline d'ingénierie. Elle lie les concepts introduits (contexte, processus, activité et méthode) avec les concepts d'architecture de l'ISO 42010 [8] et l'ISO 15288 [35].

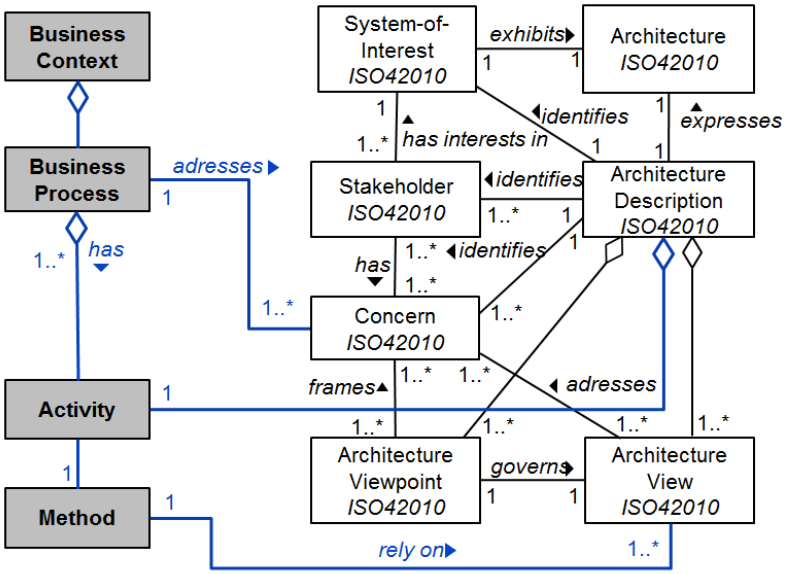


Figure 14 Relation entre le contexte d'une discipline d'ingénierie et les modèles d'architecture

Le contexte d'une discipline d'ingénierie décrit un processus qui répond à une ou plusieurs préoccupations. Le processus contient plusieurs activités qui s'appuient sur des descriptions d'architecture. Chaque activité applique une méthode qui utilise ou produit une ou plusieurs vue(s) d'architecture. Plusieurs points de vue peuvent être utilisés pour une méthode, une méthode n'est pas spécifiquement dédiée à une activité.

En complément du métamodèle, une représentation graphique est proposée pour illustrer une déclinaison possible du métamodèle. Pour représenter les processus et les activités du contexte, le langage BPMN [27] a été utilisé. Les autres formalismes n'ont pas été testés. Nous ne faisons aucune recommandation d'usage d'un formalisme plutôt qu'un autre.

La Figure 15 montre deux exemples de vues partielles, représentant chacune un contexte.

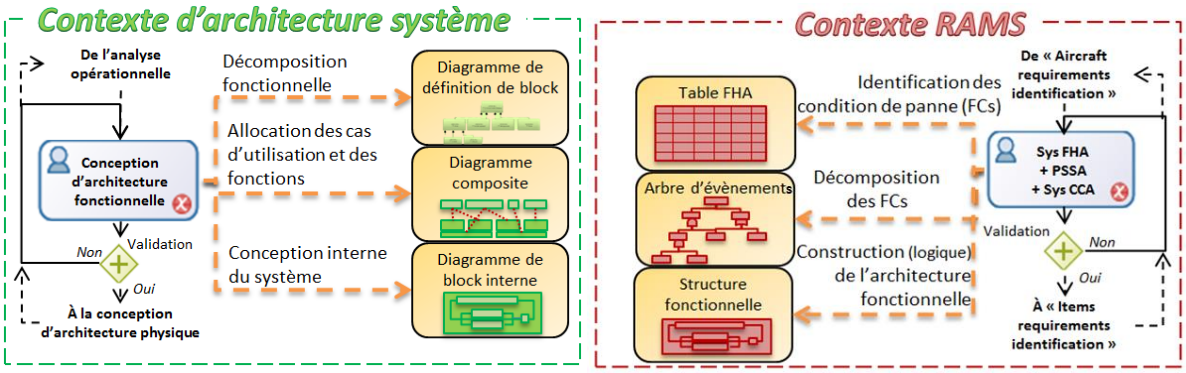


Figure 15 Exemple partiel de modèles de contexte, respectivement « Architecture système » et « Sûreté de fonctionnement »

Pour respecter la séparation des préoccupations, il est fortement déconseillé d'effectuer les interactions multidisciplinaires pendant la réalisation des activités des contextes respectifs. Elles peuvent être effectuées avant ou après le déroulement d'activités.

Un profil UML a été implémenté correspondant à la modélisation des contextes des disciplines d'ingénierie. Les détails de cette implémentation sont présentés au Chapitre VI1.3.

2.3.3. METAMODELE D'UN BESOIN DE SYNCHRONISATION

Après avoir défini et construit les modèles des contextes des disciplines d'ingénierie, le responsable de synchronisation a une meilleure compréhension des préoccupations et des pratiques. Il peut maintenant réfléchir et tenter de définir des relations entre les vues du système ayant des concepts communs ou proches. Il identifie les besoins de synchronisation.

En respectant les deux derniers métamodèles, le métamodèle d'un besoin de synchronisation est présenté dans la Figure 16.

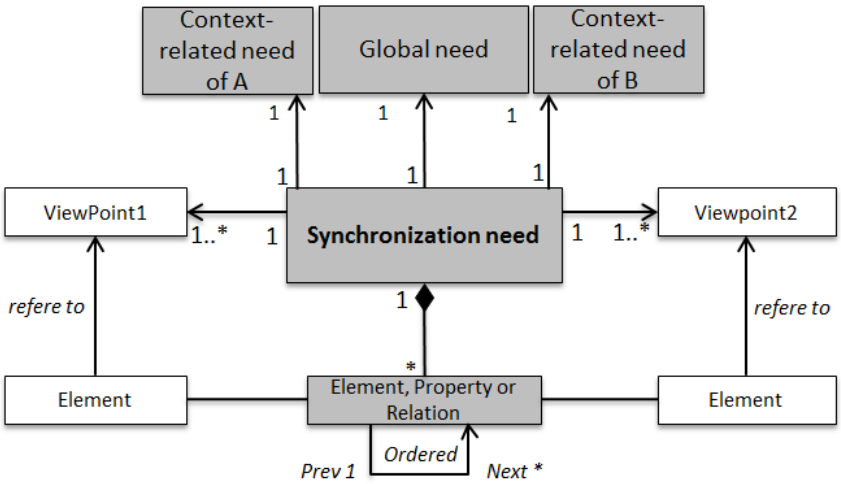


Figure 16 Métamodèle du besoin de synchronisation

Le métamodèle met en avant des questions auxquelles le responsable de synchronisation doit apporter des réponses pour chaque besoin :

- Quels sont les besoins d'une discipline vis-à-vis des autres ? (*"Global need"*, *"Context-related need of A"*, *"Context-related need of B"*)

- Quand a-t-on besoin d'échanges (à quels moments dans les processus, en amont ou en aval de la méthode,…) ? (*"Viewpoint 1"* associé au contexte A, et *"Viewpoint 2"* associé au contexte B)
- De quoi va-t-on discuter (quels objets, propriétés) ? (*"Element, Property or Relation"*)

L'objectif de ces questions est de définir, de manière informelle d'abord, des besoins de synchronisation et d'évaluer leur pertinence.
Une vue spécifique peut être construite en respect avec le métamodèle du besoin de synchronisation. Cette vue peut être présentée sous la forme d'un tableau à deux dimensions (pour considérer n disciplines d'ingénierie, $n \in \mathbb{N}^*$) étant donné que tous les éléments du métamodèle ont une multiplicité de 1 avec le besoin de synchronisation.

Le tableau (Table 6) est proposé pour répondre à ces questions et caractériser les besoins de synchronisation.

Table 6 Tableau de caractérisation des besoins de synchronisation

Nom	Besoin de la discipline A vis-à-vis de la discipline B	Besoin de la discipline B vis-à-vis de la discipline A	Raison de ce besoin	Points de vue concernés par cette interaction (2)	Les éléments, propriétés et/ou relations (2)
Besoin de synchronisation 1	*<Texte informel>*	*<Texte informel>*	*<Texte informel>*	*<Contexte><Processus><Activité><Méthode><Point de vue>*	*<Objets>*
				<Contexte><Processus><Activité><Méthode><Point de vue>	*<Objets>*
…					

En reprenant, les vues des contextes des disciplines d'ingénierie, il est possible de représenter graphiquement les besoins de synchronisation sous la forme de relations entre points de vue (Figure 17).

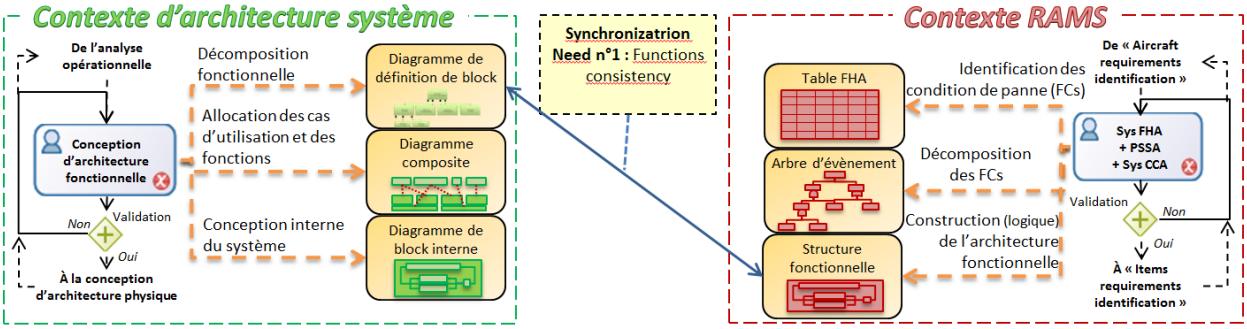


Figure 17 Contextualisation des besoins de synchronisation

Un profil UML [26] a été développé pour modéliser les relations entre les points de vue des disciplines. Les détails de cette implémentation seront présentés au Chapitre VI1.3.

Il peut être particulièrement intéressant de choisir les jalons préexistants dans les processus respectifs. La cohérence des modèles est une étape de validation qui peut être considérée parmi les étapes clés d'un projet (étape de démarrage, décision, revue, audit, …).

Voici quelques exemples de documents fournis lors d'étapes clés des processus : Cahier des Charges, Document de spécifications, Document de conception préliminaire, Documents de conception détaillée, Spécification technique du besoin (STB), Dossier de définition (DD), Dossier de justification de définition (DJD), Dossier de réalisation (DR).

Ces documents sont définis par les normes suivantes :

- NF EN 1325-1 [107] (Vocabulaire du management de la valeur, de l'analyse de la valeur et de l'analyse fonctionnelle) ;
- NFX 50 100 [108] (Analyse fonctionnelle- Caractéristiques fondamentales) ;
- ISO 21 351 [109] (Systèmes spatiaux, cahier des charges fonctionnel et Spécification de besoin).

3. DEFINITION DES CONCEPTS D'ARCHITECTURE

Pour formaliser les interactions multidisciplinaires à partir des besoins de synchronisation, il faut être en capacité de définir les niveaux d'abstraction des modèles pivots à tout moment dans les processus. Pour cela, cette section tente de caractériser le contenu des modèles utile à la représentation de l'architecture du système étudié. Elle présente donc les concepts d'élément et de relation de structuration des langages qui permettent de capturer l'architecture d'un système.

Cette formalisation permettra, dans la section 4., de définir les formalismes pivots qui permettront d'abstraire les points de vue des disciplines d'ingénierie et de les comparer.

3.1. CONCEPTS PIVOTS DE MODELISATION D'ARCHITECTURE

Les concepts réutilisés sont :

- « Architecture View », « Architecture Viewpoint », « Architecture », « Architecture Description », Préoccupation, Système (cf. 2.1 et 2.2) ;
- Partie prenantes, Modèle (cf. Chapitre I1.2 et Chapitre I1.3).

Concept(s) pivot(s) d'architecture *: c'est l'ensemble des concepts structurants communs aux modèles utilisés par les disciplines d'ingénierie. Ils font référence aux sujets soumis à la mise en cohérence lors d'une interaction multidisciplinaire. Ils définissent le métamodèle pivot qui sera utilisé lors d'une itération.*

Métamodèle pivot *: C'est la définition des modèles utilisés pour exprimer les concepts communs des vues des disciplines d'ingénierie. Il peut y avoir plusieurs métamodèles pivots dans l'exécution des interactions multidisciplinaires.*

Elément *: C'est une représentation d'un objet dans un modèle. C'est aussi un conteneur, i.e. un package pouvant contenir d'autres éléments ou du comportement, e.g. une fonction, un composant, etc. Les langages proposent des méta-classes telles que : une classe UML, un block SysML, une node en Altarica dataflow, une classe ou un block en Altarica 3.0, etc.*

Relation de structuration *: C'est une relation, i.e. une propriété des modèles qui relie plusieurs éléments entre eux. Elle a une sémantique et elle peut être de plusieurs types. Elle caractérise la structure du modèle qui reflète l'architecture du système étudié.*

3.2. MODELISATION DES CONCEPTS

3.2.1. DECLINAISONS DES ELEMENTS

Les éléments rencontrés, au cours du processus de conception ISO 15288 [35], varient de nature (cf. Chapitre I1.3), e.g. évènements, scénarios, cas d'utilisation, fonctions, composants physiques, composants logiciels, etc. Selon le système, les activités menées dans les processus et le niveau d'abstraction des vues, la nature des éléments est différente. Avec l'état des connaissances actuelles, il n'est pas possible de faire une généralisation de la déclinaison des éléments.

La Figure 18 donne un exemple de déclinaisons des éléments issues de deux contextes de modélisation (Architecture système, SEBok [14] et Analyse de sûreté de fonctionnement, ARP 4761 [21]). Cet exemple est non exhaustif, il présente un cas possible qui nécessite un raffinement.

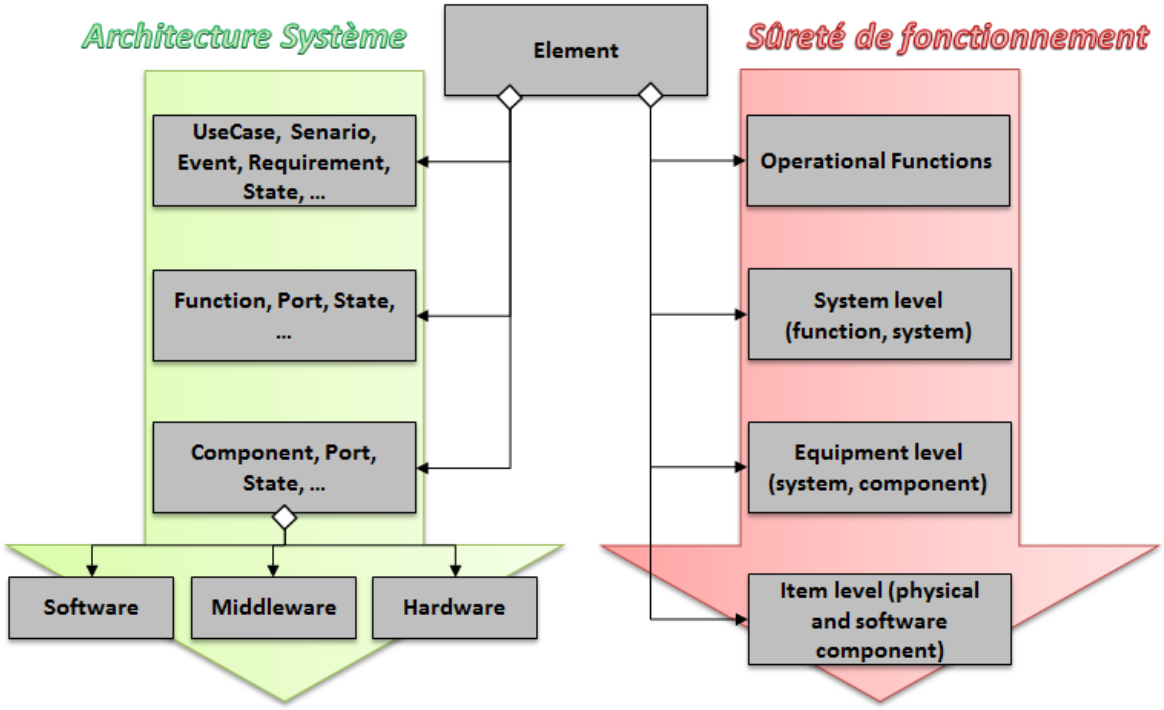


Figure 18 Déclinaison des éléments des modèles durant la conception

La typologie des éléments doit être spécifique aux activités des processus et des besoins de l'entreprise, i.e. durant les étapes très amont, les types des éléments se rapprochent de ceux du « Besoin », des « Exigences », d'« Etat », « Evènement » et « Cas d'utilisation ».

Durant les premières étapes de conception, les éléments manipulés sont d'ordre fonctionnel. On emploie souvent les déclinaisons suivantes : fonction principale, fonction secondaire, fonction contrainte, … [57], [110].

Enfin dans les dernières étapes de conception, les éléments manipulés sont des composants qui se déclinent selon des technologies, ou d'autres critères. La déclinaison que l'on retrouve le plus couramment est : Hardware, Software, Middleware, Humanware.

3.2.2. RELATIONS DE STRUCTURATION

Les relations de structuration [111] sont les propriétés utilisées pour l'organisation des modèles d'architecture système. La plupart des paradigmes de structuration (définis implicitement dans les langages) permettent la mise en relation des éléments d'un système. Elles sont effectuées par un petit ensemble de relations :

- **La relation d'agrégation des éléments** : Elle s'appuie sur l'utilisation d'un élément ou l'exploitation de ses ressources par un autre élément. Elle met en avant les interactions et les communications faites entre des éléments. Un élément peut être utilisé par un élément sans pour autant être contenu dans cet élément. Si l'un des éléments est détruit, l'autre élément existe toujours.

- **La relation de composition des éléments** (cf. Figure 19) : C'est une relation d'agrégation particulière portant une sémantique plus stricte. Elle définit les éléments « enfants » contenus dans un élément « parent ». Ce concept est utilisé à tous les niveaux d'abstraction (cf. l'ISO 15288 [35] partie 5.2.2. System structure). Si l'élément parent est détruit, alors tous les éléments enfants sont détruits.

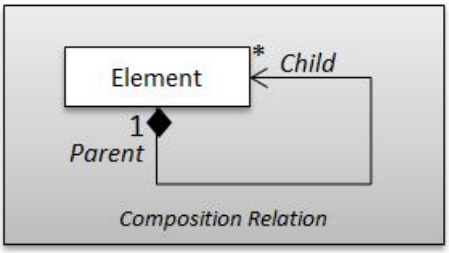


Figure 19 Modélisation du concept de composition d'élément

- **La relation d'héritage des éléments** (cf. Figure 20) : Elle permet à un élément d'obtenir les caractéristiques (propriétés) d'un autre élément. Exemple, une voiture hérite de la fonction « rouler sur la route ».

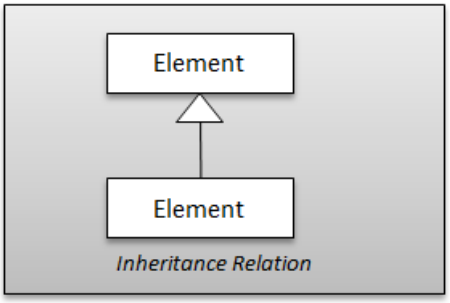


Figure 20 Modélisation du concept d'héritage

- **La relation de connexion des éléments** (cf. Figure 21) **:** Elle définit une dépendance entre les des éléments. Elle peut concerner : leur comportement, une communication entre éléments, une assertion, une association, etc.

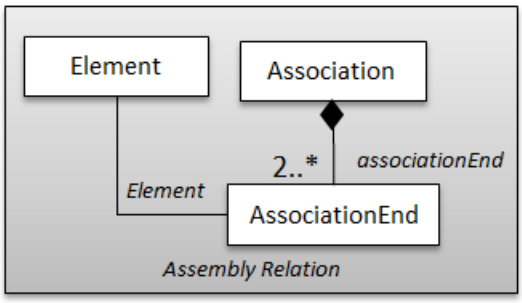


Figure 21 Modélisation du concept d'association

Les relations de structuration sont déclinées de manière plus spécifique dans les langages de modélisation. En fonction du sujet traité par la mise en cohérence durant une interaction multidisciplinaire, les concepts pivots d'architecture devront être raffinés le plus précisément possible. Le métamodèle pivot définit le niveau d'abstraction de l'interaction effectuée par les ingénieurs. Il faut donc le définir judicieusement à partir des concepts présentés ci-dessus.

Les métamodèles sont ensuite déposés dans une bibliothèque. Ils seront réutilisés pour la configuration des interactions multidisciplinaires.

3.2.3. METAMODELE PIVOT, UNE COMBINAISON D'ELEMENTS ET DE RELATIONS DE STRUCTURATION

Un concept pivot d'architecture (Cpa) est une combinaison d'un ensemble de relations de structuration (R_{struc}) et d'un ensemble de types d'élément (E_{type}), pour un objectif d'étude particulier ($ObjeE$). Il capitalise les concepts prisent en compte par une application de la synchronisation.

$ObjeE$ = objectif d'étude, $ObjeE$ = paramètre du Cpa

R_{struc} = Relations de structuration considérés, $R_{struc} = \{R_{struc\,1}, \dots, R_{struc\,n}\}$, avec $n \in \mathbb{N}^*$

E_{type} = Type des éléments considérés, $E_{type} = \{E_{type\,1}, \dots, E_{type\,m}\}$, avec $m \in \mathbb{N}^*$

$Cpa = concept\ pivot\ d'architecture$, $Cpa_{ObjetE} \subseteq R_{struc} \times E_{type}$

L'ensemble E_{type}peut, par exemple, représenter des éléments présentés à la section 3.2.1. De même, l'ensemble R_{struc}peut représenter des relations présentées à la section 3.2.2.

Plusieurs exemples simples de concepts pivots d'architecture sont présentés ci-dessous, Table 7, Table 8, Table 9, Table 10. Les combinaisons $R_{struc} \times E_{type}$ peuvent être plus sophistiquées si le responsable de synchronisation le souhaite.

Table 7 Exemple 1 de concept pivot d'architecture pour la mise en cohérence

Exemple 1 :	**Terme**	**Description**
	Cpa	Un élément possède un nom
	R_{struc}	Relation de composition
	E_{type}	Tout type d'élément
	$ObjeE$	Représenter les éléments d'un système en conception d'architecture

La Figure 23 représente le métamodèle associé au concept pivot d'architecture de l'exemple 1.

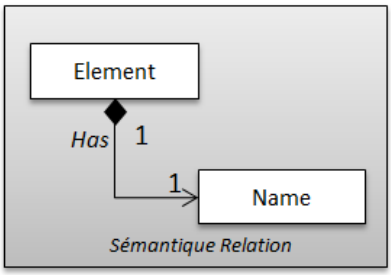


Figure 22 Métamodèle d'un élément possédant un nom

Table 8 Exemple 2 de concept pivot d'architecture pour la mise en cohérence

Exemple 2 :	**Terme**	**Description**
	Cpa	Allocation des cas d'utilisation sur une fonction
	R_{struc}	Relation d'héritage
	E_{type}	Cas d'utilisation comme élément sommet Fonction comme élément feuille
	$ObjeE$	Représenter l'allocation des propriétés d'un cas d'utilisation à une fonction en conception d'architecture.

La Figure 23 représente le métamodèle associé au concept pivot d'architecture de l'exemple 2.

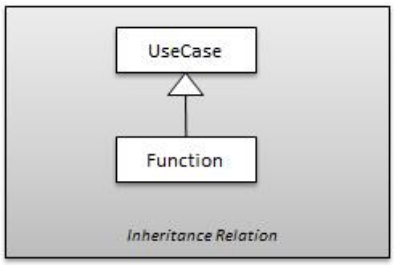


Figure 23 Métamodèle de la relation d'héritage

Table 9 Exemple 3 de concept pivot d'architecture pour la mise en cohérence

Exemple 3 :	**Terme**	**Description**
	Cpa	Décomposer une fonction en sous-fonctions
	R_{struc}	Relation de composition
	E_{type}	Les éléments sont des fonctions
	$ObjeE$	Modéliser la hiérarchie des fonctions d'un système en sûreté de fonctionnement.

La Figure 24 représente le métamodèle associé au concept pivot d'architecture de l'exemple 3.

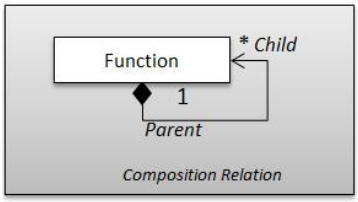


Figure 24 Métamodèle de la relation de composition

Table 10 Exemple 4 de concept pivot d'architecture pour la mise en cohérence

Exemple 4 :	**Terme**	**Description**
	Cpa	Association des composants physiques d'un système
	R_{struc}	Relation de connexion
	E_{type}	Les éléments connectés sont des « composants physiques » et les éléments de connexion sont des « associations »
	$ObjeE$	Modéliser les connexions entres les entrées et les sorties des composants physiques en sûreté de fonctionnement.

La Figure 25 représente le métamodèle associé au concept pivot d'architecture de l'exemple 4.

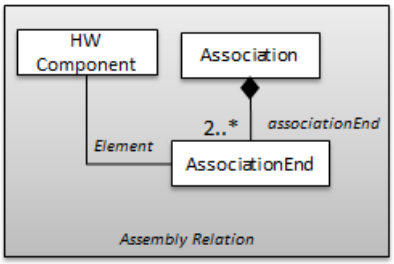


Figure 25 Métamodèle de la relation de connexion

Une bibliothèque peut être définie pour recenser des concepts pivots d'architecture. Elle est utilisée lors de la configuration de synchronisation de modèles.

3.2.4. ORDONNANCEMENT DES CONCEPTS PIVOTS D'ARCHITECTURE

Comme les éléments et les relations de structuration se déclinent dans un certain ordre dans les processus, les concepts pivots d'architecture aussi. Il est important de caractériser l'ordre dans lequel ils seront traités par la synchronisation de modèles pour éviter des incohérences impossibles à résoudre.

Il est possible de définir des contraintes pour évaluer la faisabilité d'exécution des interactions multidisciplinaires. Plusieurs contraintes génériques sont définies, ci-dessous, à titre d'exemple.

Contrainte 1 : Les éléments traités lors d'interactions multidisciplinaires doivent respecter l'ordre suivant :

Cas d'utilisation < Fonction < Composant < Elément d'interface (e.g. input, output)

Contrainte 2 : Les relations de structuration traités lors d'interactions multidisciplinaires doivent respecter l'ordre suivant :

Relations de composition < Relations d'héritage < Relations d'agrégation < Relation de connexion

Contrainte 3 : Les concepts pivots d'architecture traitées lors d'interactions multidisciplinaires au niveau d'une architecture physique doivent respecter l'ordre des concepts suivant :

Composant < Relations de composition < Elément d'interface (e.g. input, output) < Relations de connexion

4. CONFIGURATION DES INTERACTIONS MULTIDISCIPLINAIRES

Pour appliquer la synchronisation de modèles, il est nécessaire de configurer chaque interaction multidisciplinaire identifiée par les besoins de synchronisation. La configuration introduit le concept de point de synchronisation qui formalise les vues, les abstractions, la comparaison et les concrétisations possibles qui seront effectuées par les disciplines pendant la conception du système.

La section 4 présente la configuration des interactions multidisciplinaires en détaillant :

- les concepts associés au « point de synchronisation » ;
- les métamodèles associant les concepts ;
- des représentations graphiques pour caractériser les points de synchronisation et organiser leur ordonnancement.

C'est le responsable de synchronisation qui effectuera cette étape pour spécifier en détail l'exécution des synchronisations tout au long du cycle de développement.

4.1. CONCEPTS D'UN POINT DE SYNCHRONISATION

Point de synchronisation : *c'est une configuration de mise en cohérence de deux points de vue caractérisant trois fonctions de base (l'abstraction, la comparaison et la concrétisation). Il peut avoir des antécédents.*

Un point de synchronisation est composé de 4 ensembles de concepts : les concepts pivots d'architecture, les mappings, les listes compromis et les conditions de validation de la cohérence.

La notion de **Concept pivot d'architecture** est définie à la section 3.1

L'approche mise en œuvre encourage l'utilisation de cycles itératifs pour la synchronisation. Un concept pivot d'architecture est une sous partie implicite des paradigmes de structuration des modèles et des points de vue ciblés. Il définit sur quoi porte la mise en cohérence qui sera établie.

Mapping *: le mapping est un ensemble ordonné de relations reliant les objets du ou des point(s) de vue d'un contexte vers les objets du métamodèle pivot. La notion de mapping est fortement liée aux principes de transformation de modèles (cf. 6.3 Transformation de modèles).*

Un point de synchronisation contient autant de mappings que de contextes considérés.

Listes de compromis : *ce sont des ensembles d'opérations pouvant être appliqués sur les modèles. Chaque discipline d'ingénierie peut proposer d'appliquer un compromis pour tenter de résoudre une incohérence identifiée. Les listes peuvent être limitées par le rôle des disciplines.*

Un point de synchronisation contient autant de listes de compromis que de contextes considérés.

Conditions de validation*: c'est l'ensemble de règles de validation qui doivent être vérifiées avant chaque application de la synchronisation.*

Par défaut, des opérations génériques sont : ajouter, modifier, supprimer un objet, renommer une propriété, déplacer un élément, ajouter un commentaire sur un objet, etc. Elles peuvent également être plus riches comme des patterns de structuration. Par exemple « appliquer une

redondance ». C'est probablement la barrière de sûreté la plus utilisée. Ces opérations se concrétisent différemment selon le contexte dans lequel elles se déclinent.

Des bibliothèques peuvent être établies pour chaque concept introduit, cependant elles resteront spécifiques aux contextes considérés.

4.2. MODELISATION DES CONCEPTS

4.2.1. METAMODELE D'UN POINT DE SYNCHRONISATION

La Figure 26, illustre le métamodèle d'un point de synchronisation. Elle lie les concepts introduits (concept pivot d'architecture, Mapping, Liste de compromis et condition de validation) avec les contextes des disciplines.

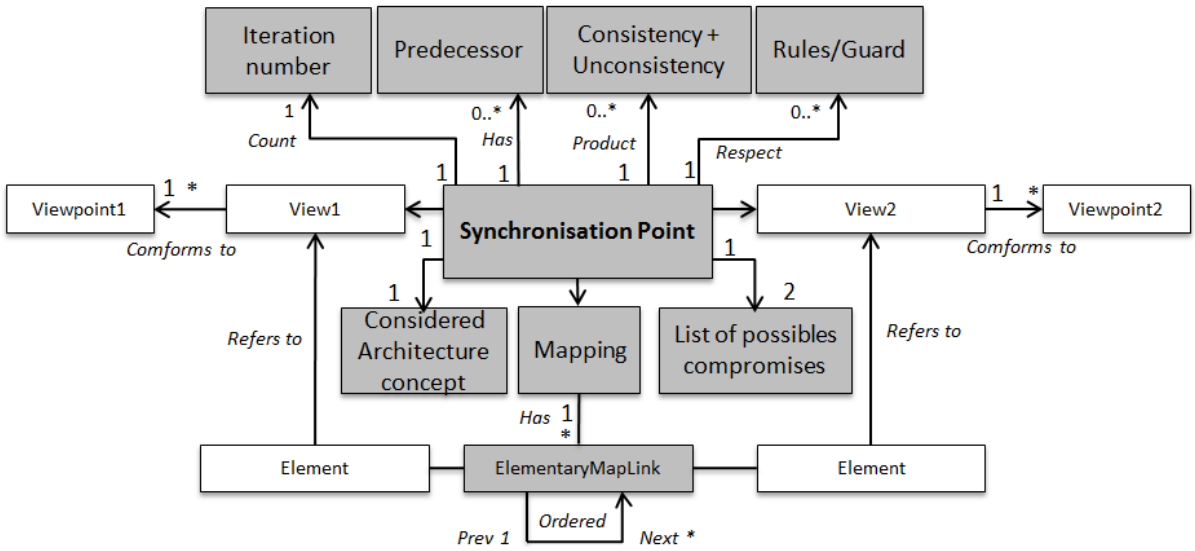


Figure 26 Métamodèle du point de synchronisation

La définition des points de synchronisation est issue des besoins de synchronisation identifiés.

4.2.2. DEPENDANCE DU BESOIN ET DES POINTS DE SYNCHRONISATION

A partir du besoin de synchronisation, un ou plusieurs points de synchronisation peuvent être définis. Cela dépend du raffinement des concepts pivots d'architecture définis.

Le responsable de synchronisation peut choisir des concepts pivots d'architecture très simples afin de séparer les préoccupations lors d'interactions. Ceci permet de faciliter la discussion entre les ingénieurs, cependant elle multiplie le nombre d'interactions.

La Figure 27 présente les dépendances entre le besoin et le point de synchronisation.

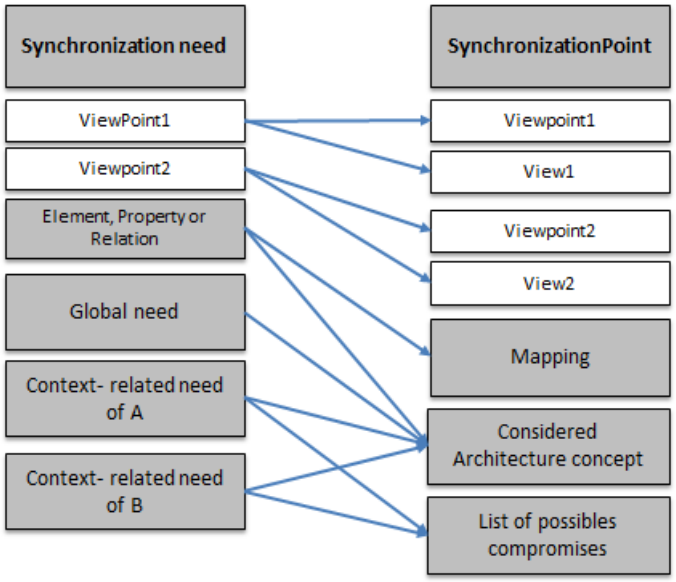


Figure 27 Dépendances entre les concepts de besoin de synchronisation et de point de synchronisation

4.2.3. ORDONNANCEMENT DES POINTS DE SYNCHRONISATION

Les points de synchronisation dépendent les uns des autres selon :

- le raffinement et les dépendances des concepts pivots d'architecture ;
- les processus des contextes des disciplines d'ingénierie ;
- le type d'objet traité (opérationnel, fonctionnel, composant physique, etc.).

L'application de la synchronisation est une démarche itérative. Pour chaque itération, un modèle de cohérence est produit, i.e. un ensemble de relations de cohérence et d'incohérences. Ceci sera détaillé dans la section suivante.

Ainsi, un point de synchronisation est appliqué autant de fois que nécessaire, i.e. que plusieurs itérations seront effectuées jusqu'à ce que les correspondances entre les vues soient entièrement satisfaites.

C'est pourquoi l'ordonnancement respecte les conditions de validation suivantes :

- il ne doit pas violer l'ordre des concepts de structuration des paradigmes (cf. 3.2.4) ;
- il ne doit pas violer l'ordre des activités des processus concernés ;
- il ne doit pas violer l'ordre des niveaux d'abstraction des objets (cf. 3.2.1).

Pour autoriser ou non l'application d'une interaction, i.e. l'exécution d'un point de synchronisation et de ses itérations, plusieurs règles sont définies dans la Figure 28 et la Figure 29. Elles sont vérifiées avant chaque exécution.

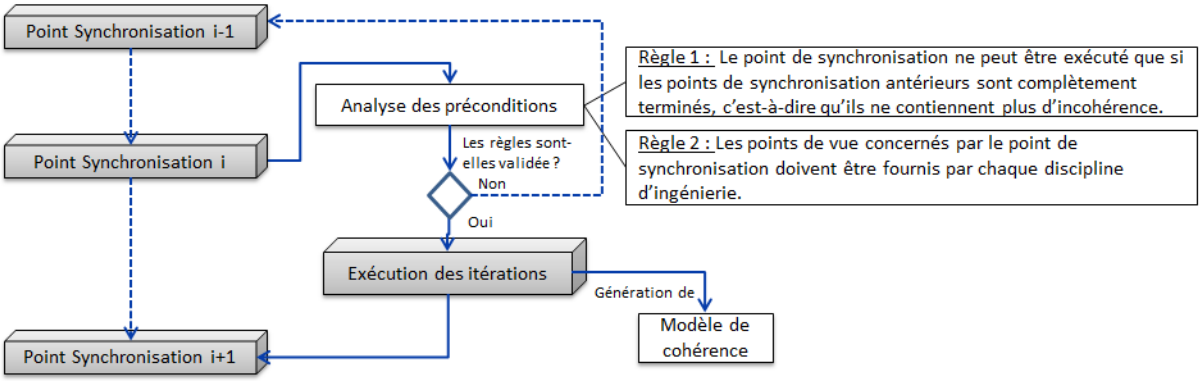


Figure 28 Processus d'exécution d'un point de synchronisation

Si l'une des règles 1 ou 2 n'est pas validée, l'exécution ne sera pas autorisée. Ces règles pré-conditionnent également l'exécution des itérations. Une règle additionnelle, règle 4, non bloquante est définie :

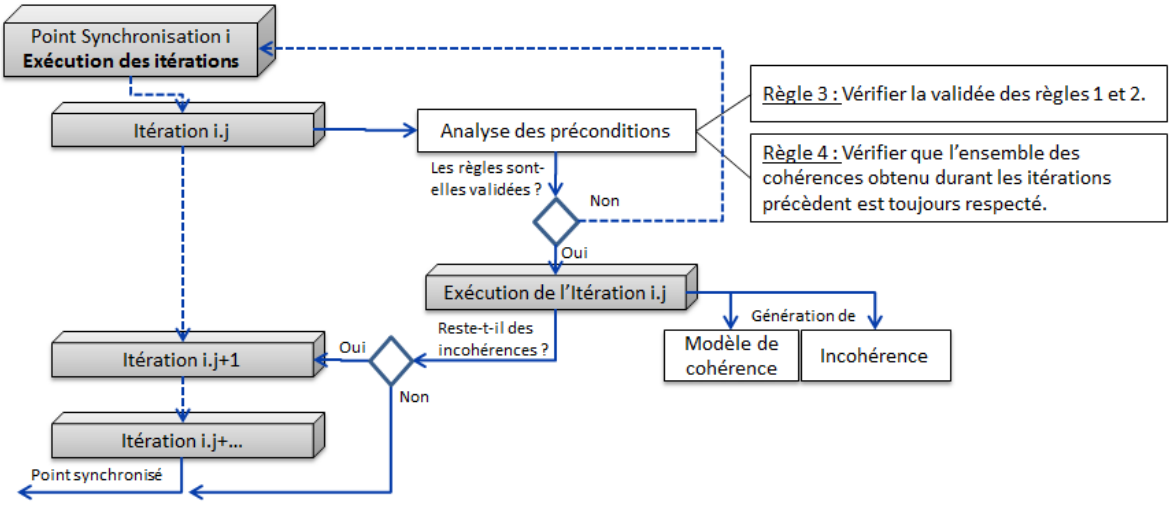


Figure 29 Processus d'exécution d'itération

Si la règle 3 n'est pas vérifiée, cela implique que l'itération ne peut pas être effectuée. Il sera nécessaire de revenir sur un certain point de synchronisation antérieur.

Si la règle 4 n'est pas vérifiée, alors au moins une relation de cohérence identifiée durant une itération précédente (du même point de synchronisation) n'est plus valide. L'itération pourra s'exécuter mais elle devra comparer des éléments déjà traités.

Pour valider ces règles et construire des modèles de cohérence, il faut définir ce que sont une cohérence et une incohérence.

5. MISE EN COHERENCE

L'application de la synchronisation de modèles exécute des points de synchronisation par itération. Chacun d'eux construit un modèle de cohérence contenant un ensemble de relations de cohérences et un ensemble d'incohérences.

Un modèle de cohérence est le résultat d'une comparaison entre plusieurs vues abstraites selon certains concepts d'architecture. Il est construit par les disciplines d'ingénierie elles-mêmes. C'est pourquoi une sémantique et une syntaxe graphique sont définies. Elles

permettront d'aider les ingénieurs à saisir les relations entre les éléments de leur vue qui caractérisent la cohérence.

5.1. CONCEPTS DE MISE EN COHERENCE

La mise en cohérence est une activité qui permet de caractériser des processus. Ici, la mise en cohérence qualifie le processus de synchronisation de modèles, i.e. l'art de rendre les modèles équivalents, même projetés. Elle induit une seconde notion qui sera présentée à la section 7, c'est la traçabilité des modèles.

La cohérence est une propriété de la mise en cohérence. Elle est rattachée à une relation entre les contenus de modèles.

Une cohérence *est une propriété d'une relation entre des éléments de différents contextes. C'est une équivalence en rapport à un certain sujet, i.e. que les éléments font référence à une même réalité. Elle peut être complètement ou partiellement satisfaite.*

Une incohérence *est la non-satisfaction d'une relation de cohérence. Elle indique un écart d'interprétation entre les vues.*

On définit $R_{coh\ (\beta,C)}$, la relation de cohérence entre des éléments ou des relations de plusieurs contextes. Elle est définie par l'ensemble de ses extrémités.

$$Soit\ \beta = la\ cohérence\ et\ \mathrm{C} = \{\mathrm{C}_1, \dots, \mathrm{C}_n\}\ l'\text{ensemble des contextes, avec n } \in \mathbb{N}^*$$

$$\beta\ et\ C\ sont\ les\ deux\ paramètres\ de\ la\ relation\ R_{coh}$$

$$R_{coh\ (\beta,C)} \subseteq \bigcup_{i=1}^{n} \mathrm{C}_i \times\ ]0,1]$$

$$R_{coh\ (\beta,C)} = \{\mathrm{Re}_1, \dots, Re_m\}, avec\ m\ le\ nombre\ d'extrémité, m \in \mathbb{N}^+\ et\ m \geq 2$$

$$\mathrm{Re}_i\ est\ l'extrémité\ i\ de\ la\ relation\ R_{coh\ (\beta,C)}, avec\ i\ \in]1, m]$$

On définit Re_i, une des extrémités d'une relation $R_{coh\ (\beta,C)}$.

$$Soit\ un\ Element\ \in \bigcup_{i=1}^{n} \mathrm{C}_i\ \text{avec n } \in \mathbb{N}^*$$

$$et\ x\ \text{la complétude d'un Element selon } \beta, x \in\ \mathbb{R} \text{ et } x \in\ ]0,1]$$

$$\forall\ i\ \in]1, m], \qquad le\ couple\ \mathrm{Re}_i = (Element, x)\ \ désigne\ l'extrémité\ d'une\ relation\ \ R_{coh\ (\beta,C)}$$

La complétude est une valeur comprise dans l'intervalle]0,1]. Elle est subjective car très difficilement mesurable. Des paliers peuvent être définis pour caractériser la complétude et être ramenés à des valeurs arbitraires comprises entre 0 et 1.

Remarque : une extrémité où la complétude est nulle n'a aucun sens.

Une relation de cohérence est complète si et seulement si la somme des Re_i de chaque contexte est égale à 1, sinon elle est incomplète.

5.2. REPRESENTATION GRAPHIQUE D'UNE RELATION DE COHERENCE

Dans les premiers exemples de cette section, nous avons représentées graphiquement la sémantique des éléments par des « patatoïdes » pour aider le lecteur à comprendre la signification des relations de cohérence. Dans la réalité, elles ne sont jamais représentées. Ainsi un élément E_1, qui porte la sémantique S_1 sera représenté comme dans la Figure 30.

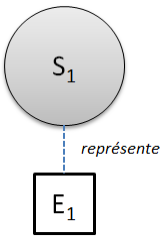
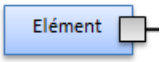
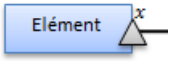


Figure 30 Représentation d'un élément et de sa sémantique

La représentation graphique d'une relation de cohérence est composée :

- d'une droite (———) représentant la relation $R_{coh\,(\beta,C)}$ reliée aux extrémités Re_i ;
- de formes représentant les extrémités Re_i et leur complétude x. Elles sont rattachées à une relation $R_{coh\,(\beta,C)}$ et un élément d'un contexte C.
    - Si la complétude est de 1, alors l'extrémité Re_i est représentée par un carré 

.
    - Si la complétude est comprise entre $]0,1[$, alors l'extrémité est représentée par un triangle et la valeur de x 

.

La Figure 31 illustre un exemple concret, de relations de cohérence entre des éléments fonctionnels d'une voiture.

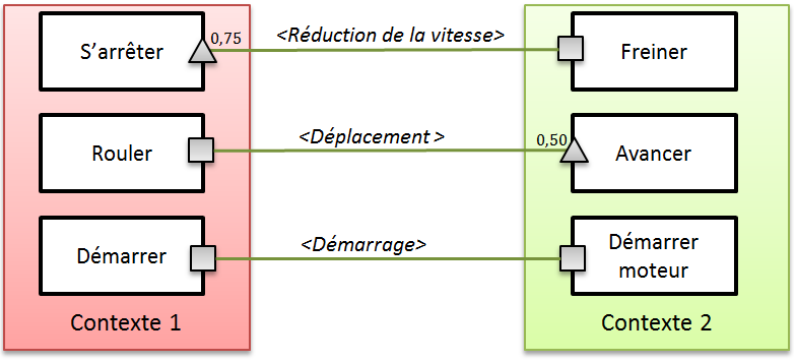


Figure 31 Exemple de trois relations de cohérence

Première relation de cohérence : $R_{coh\,(Réduction\ de\ la\ vitesse,12)} = \{ Re_1, Re_2 \}$
$Re_1 =< S'arrêter, 0.75 > et\ Re_2 =< Freiner, 1 >$

Deuxième relation de cohérence : $R_{coh\,(Déplacement,12)} = \{ Re_3, Re_4 \}$
$Re_3 =< Rouler, 1 > et\ Re_4 =< Avancer, 0.5 >$

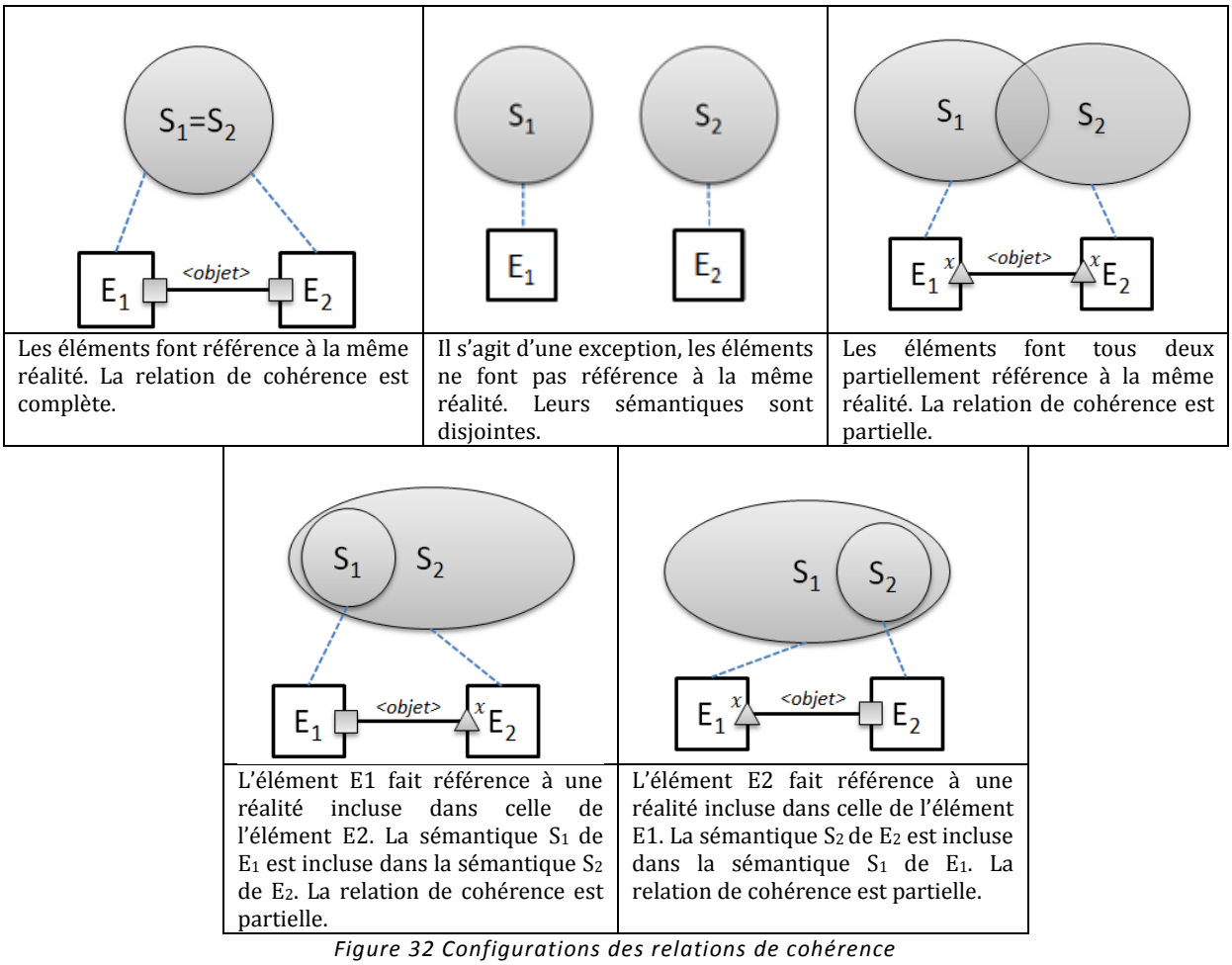
Troisième relation de cohérence : $R_{coh\,(Démarrage,12)} = \{ Re_5, Re_6 \}$
$Re_5 =< Démarrer, 1 > et\ Re_6 =< Démarrer\ moteur, 1 >$

La première relation de cohérence $R_{coh\ (Réduction\ de\ la\ vitesse,12)}$ n'est pas complètement satisfaite, car le sens de s'arrêter est plus fort que freiner. Les ingénieurs ont défini 0.75 de complétude arbitrairement pour signifier que l'écart des fonctions n'est pas très grand. En effet, s'arrêter implique qu'on est en train de freiner.

La seconde relation de cohérence $R_{coh\ (Déplacement,12)}$ n'est pas complètement satisfaite, car le sens de « avancer » est plus fort que rouler. Les ingénieurs ont défini 0.5 de complétude arbitrairement pour indiquer que la fonction « avancer » ne représente que la moitié de la fonction « Rouler ». En effet, on peut rouler en avançant ou en reculant.

La troisième relation de cohérence $R_{coh\ (Démarrage,12)}$ est complètement satisfaite, car le sens de démarrer et démarrer moteur font référence à la même fonction du véhicule.

Cas particulier (cf. Figure 32) **:** Lorsqu'il n'y a que deux contextes et un seul élément par contexte, les configurations possibles d'une relation de cohérence sont les suivantes :



Les éléments font référence à la même réalité. La relation de cohérence est complète.	Il s'agit d'une exception, les éléments ne font pas référence à la même réalité. Leurs sémantiques sont disjointes.	Les éléments font tous deux partiellement référence à la même réalité. La relation de cohérence est partielle.
L'élément E1 fait référence à une réalité incluse dans celle de l'élément E2. La sémantique S_1 de E_1 est incluse dans la sémantique S_2 de E_2. La relation de cohérence est partielle.	L'élément E2 fait référence à une réalité incluse dans celle de l'élément E1. La sémantique S_2 de E_2 est incluse dans la sémantique S_1 de E_1. La relation de cohérence est partielle.	

Figure 32 Configurations des relations de cohérence

Exemple particulier : Lorsqu'il n'y a que trois contextes et un seul élément par contexte, les configurations possibles d'une relation de cohérence sont les suivantes :

L'exemple, Figure 33, présente 3, 2 et 4 éléments respectivement dans les 3 contextes. Deux relations de cohérence sont représentées.

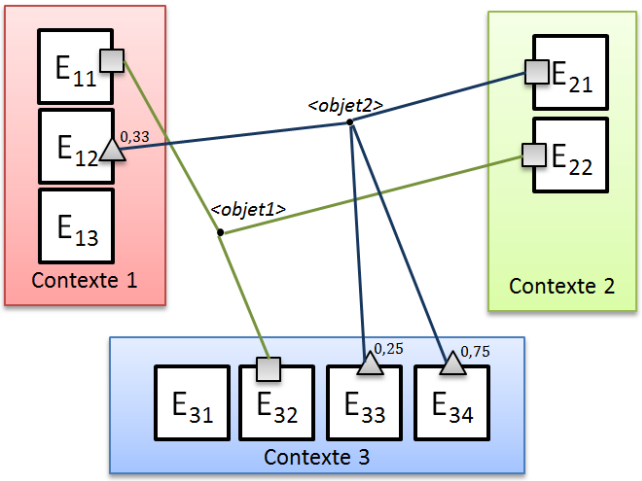


Figure 33 Exemple de deux relations de cohérence

Les deux relations de cohérence présentées, se définissent :

Première relation de cohérence : $R_{coh\,(Object1,123)} = \{ Re_1, Re_2, Re_3 \}$

$Re_1 =< E_{11}, 1 >,\ Re_2 =< E_{22}, 1 >\ et\ et\ Re_3 =< E_{32}, 1 >$

Deuxième relation de cohérence : $R_{coh\,(Object2,123)} = \{ Re_4, Re_5, Re_6, Re_7 \}$

$Re_4 =< E_{12}, 0.33 >, Re_5 =< E_{21}, 1 >, Re_6 =< E_{33}, 0.25 >\ et\ Re_7 =< E_{34}, 0.75 >$

Les relations de cohérence peuvent également concerner plusieurs concepts d'architecture. L'exemple, Figure 34, illustre deux relations de cohérence. Les composants A et B sont cohérents selon leur composition. Cependant, ils sont aussi incohérents selon la fonction qu'ils servent.

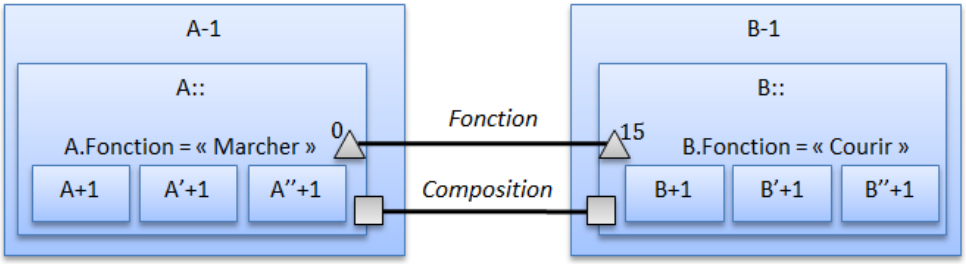


Figure 34 Exemple de relations de cohérence sur plusieurs concepts d'architecture

Les relations de cohérence qui concernent différents concepts d'architecture peuvent faire référence aux mêmes éléments.

6. APPLICATION DE LA SYNCHRONISATION DE MODELES

Cette section présente l'application de la synchronisation de modèles, en détaillant :

- les concepts d'exécution de la synchronisation ;
- le modèle et la méthode d'exécution de synchronisation des modèles ;
- les fonctions associées aux concepts de synchronisation des modèles ;
- les dépendances entre la configuration et l'application de la synchronisation ;
- les techniques satisfaisant les fonctions de synchronisation de modèles.

L'application de la synchronisation est un support de discussion pour la mise en cohérence et la résolution d'incohérences. Son exécution est manipulée par les ingénieurs et les architectes qui restent au cœur de sa mise en œuvre.

6.1. CONCEPTS DE SYNCHRONISATION

Quatre concepts sont définis pour l'exécution de la synchronisation de modèles.

Vérification des conditions de validation : *Etape de lancement d'une itération, elle vérifie si les conditions de validation sont respectées. Si l'une des règles 1, 2 ou 3 (cf. 3.2.4) n'est pas validée, elle en informera les ingénieurs et les architectes. Si la règle 4 (cf. 3.2.4) n'est pas validée, l'itération pourra commencer et les ingénieurs seront informés de la relation de cohérence perdue. Si toutes les règles sont validées, alors la vérification autorisera l'exécution de la synchronisation de modèles.*

Abstraction : *L'abstraction est une technique de transformation de modèles. C'est l'activité inverse du raffinement. Les informations d'un modèle sont regroupées ou éliminées pour simplifier le modèle. L'objectif de l'abstraction est de masquer certaines informations pour mettre l'accent sur une préoccupation spécifique. Si le modèle M1 raffine le modèle M2 alors M2 est une abstraction de M1.*

Comparaison : *La comparaison de modèles est une technique de construction de relations spécifiques entre des éléments de deux modèles. La comparaison est chargée de construire des relations et d'en interpréter des résultats selon des critères. Les modèles comparés doivent être décrits par le même langage.*

Dans notre cas, la comparaison traite de la mise en cohérence des modèles utilisés dans les processus. C'est une technique semi-automatique, car ce sont les ingénieurs et les architectes qui construisent les relations entre les éléments. La comparaison assiste les utilisateurs à l'aide de représentations graphiques puis identifie les relations de cohérence et les incohérences.

Concrétisation : *La concrétisation est une technique de transformation de modèles. La concrétisation est une activité de raffinement de modèle. Des informations sont ajoutées dans un modèle pour le raffiner. L'objectif de la concrétisation est de rendre plus concret le contenu du modèle.*

6.2. MODELISATION D'UNE APPLICATION DE SYNCHRONISATION

6.2.1. EXECUTION DE L'APPLICATION

L'application de la synchronisation suit une méthode à 5 étapes. Chaque étape tire parti de la configuration des points de synchronisation pour respecter les contextes des disciplines d'ingénierie.

C'est une méthode collaborative et itérative :

1. Vérification des relations de cohérence des points de synchronisation et des itérations précédentes ;
2. Abstraction des vues de chaque contexte ;
3. Comparaison des vues et éléments résultants de l'abstraction ;
4. Si au moins une incohérence est observée, alors :
    1. Concrétisation de compromis sur les vues d'un ou plusieurs contextes ;
    2. Modification par les vues pour satisfaire le compromis ;
5. Si aucune incohérence n'est constatée, alors la mise en cohérence des vues est validée.

Elle est exécutée plusieurs fois jusqu'à ce qu'il n'y ait plus d'incohérence identifiée. Une itération s'applique donc suivant 5 étapes successives : Vérification, Abstraction, Comparaison, Choix de concrétisation, Concrétisation, Adaptation des vues.

Le modèle d'exécution correspondant à la méthode est représenté dans la Figure 35.

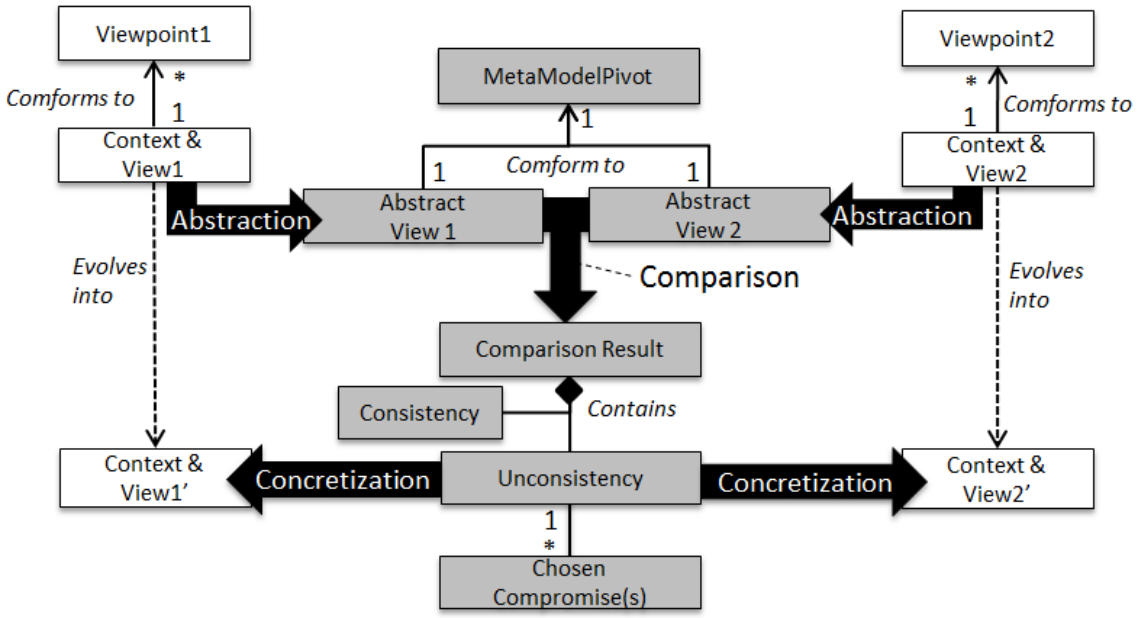


Figure 35 Modèle d'exécution des fonctions de synchronisation

6.2.2. FONCTIONS, ENTREES ET SORTIES DES ETAPES DE SYNCHRONISATION

L'exécution de la synchronisation de modèles manipule plusieurs éléments en entrées ou sorties des fonctions. On considère :

- initialement, les contextes A et B ayant respectivement la vue 1 ou 2 ;
- après l'abstraction, les vues abstraites 1 et 2, chacune issue du contexte A ou B ;
- après la comparaison, la liste des relations de cohérence et la liste des incohérences ;
- enfin, après la concrétisation, les contextes A et B ayant respectivement la vue 1' ou 2'.
- tout au long de la synchronisation, les points de synchronisation, les bibliothèques ;
- après chaque fonction, la traçabilité des étapes effectuées ;

L'ensemble des éléments sont les suivants :

- La vue 1 du contexte A ;
- Vue abstraite 1 ;
- Vue 1' du contexte A ;
- Les points de vue du contexte A ;
- La vue 2 du contexte B ;
- Vue abstraite 2 ;
- Vue 2' du contexte B ;
- Les points de vue du contexte B ;
- Les points de synchronisation (cf.4.1) ;
- Bibliothèque des concepts d'architecture (cf. 3.2.3) ;
- Bibliothèques des mappings (cf.4.1) ;
- Liste des relations de cohérence (cf.5.1) perdues ;
- Bibliothèques des compromis (cf.4.1) ;
- Liste des relations de cohérence (cf.5.1) ;
- Liste d'incohérences (cf. 5.1) ;
- Traçabilité des itérations antérieures (cf. 7) ;
- Traçabilité des points de synchronisation (cf. 7) ;
- Traçabilité des abstractions (cf. 7) ;
- Traçabilité de la comparaison (cf. 7) ;
- Traçabilité de la concrétisation (cf. 7) ;

La vérification des conditions de validation réalise les fonctions suivantes :

- Evaluation de la validité des règles. Cela concerne l'ordonnancement des points de synchronisation et les relations de cohérence déjà construites ;
- Listing des relations de cohérence perdues s'il en identifie ;
- Autorisation ou pas de l'exécution des étapes suivantes.

La Figure 36 présente les flux des données d'entrées et de sorties de la fonction « Vérification ». Les éléments d'entrées sont représentés à gauche de la fonction et les résultats produits sont à droite.

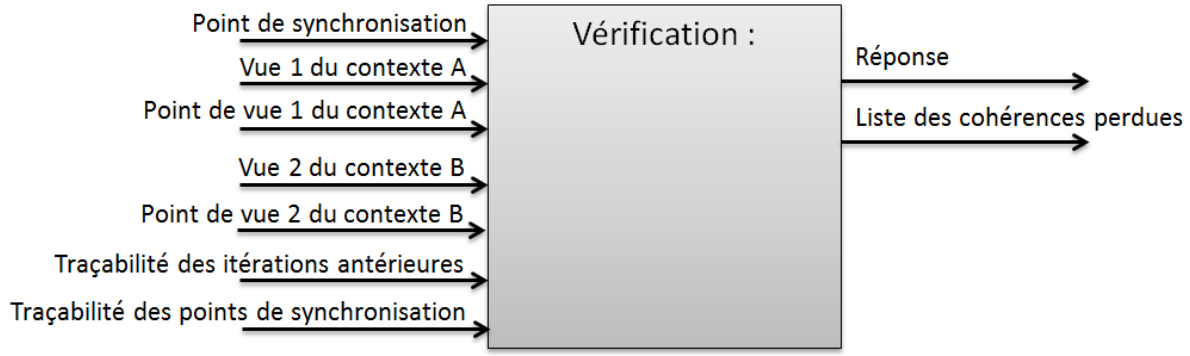


Figure 36 Fonction de vérification de l'application de synchronisation

L'abstraction réalise les fonctions suivantes :

- Exécution de transformations de modèles à partir des vues 1 et 2 des contextes A et B vers des vues abstraites ;
- Respect du métamodèle pivot selon les concepts d'architecture choisis ;
- Production des relations de traçabilité entre les éléments des vues et leurs abstractions.

La Figure 37 présente les flux des données d'entrées et de sorties de la fonction « Abstraction ». Les éléments d'entrées sont représentés à gauche de la fonction et les résultats produits sont à droite.

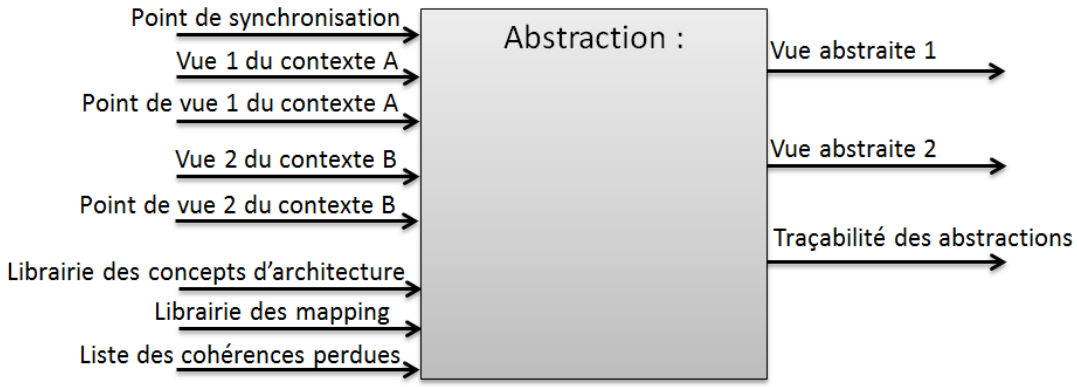


Figure 37 Fonction d'abstraction de l'application de synchronisation

La comparaison réalise les fonctions suivantes :

- Construction des représentations graphiques d'aide à la saisie des relations de cohérence entre les éléments des vues abstraites 1 et 2 ;
- Construction des relations de cohérence à partir des saisies des ingénieurs et des architectes ;
- Identification des incohérences à partir de règles de satisfaction de contraintes ;
- Construction de deux listes : des relations de cohérence et des incohérences ;
- Construction des relations de traçabilité entre les éléments des deux vues abstraites.

La Figure 38 présente les flux des données d'entrées et de sorties de la fonction « Comparaison ». Les éléments d'entrées sont représentés à gauche de la fonction et les résultats produits sont à droite.

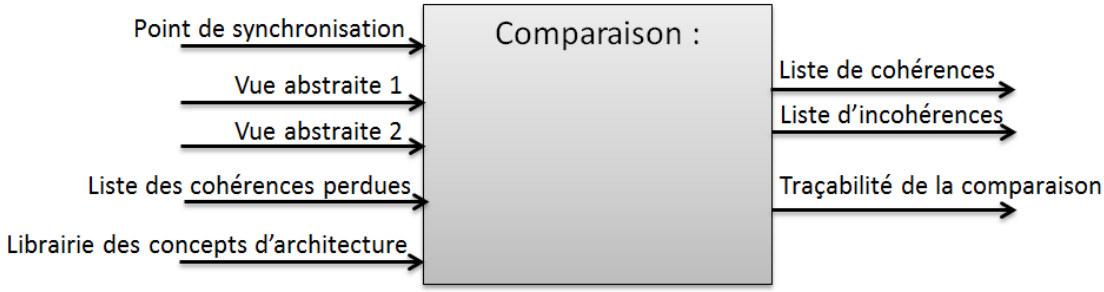


Figure 38 Fonction de comparaison de l'application de synchronisation

La concrétisation réalise les fonctions suivantes :

- Récupération de(s) compromis choisi(s) par les ingénieurs et les architectes pour résoudre le(s) incohérence(s) ;
- Construction des vues 1' et 2' des contextes A et B annotées par les compromis choisis ;
- Construction des relations de traçabilité entre l'(les) élément(s) des vues abstraites, l'(les) élément(s) des vues 1' et 2' et les compromis.

La Figure 39 présente les flux des données d'entrées et de sorties de la fonction « vérification ». Les éléments d'entrées sont représentés à gauche de la fonction et les résultats produits sont à droite.

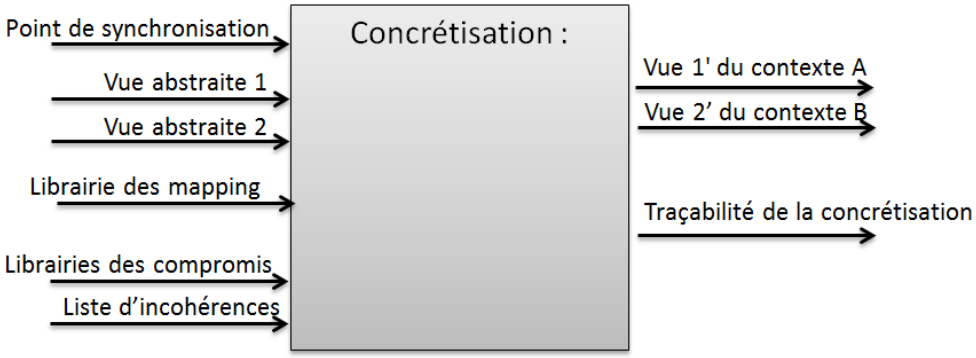


Figure 39 Fonction de concrétisation de la synchronisation

Les fonctions de synchronisation génèrent systématiquement des traces. Ces dernières seront présentées dans la section « Traçabilité et histoires des modèles ».

6.2.3. DEPENDANCE DE L'APPLICATION ET DU POINT DE SYNCHRONISATION

A partir de la configuration des interactions multidisciplinaires, notamment des points de synchronisation toutes les parties automatisables de l'exécution de la synchronisation de modèles peuvent être générées. La Figure 40 présente les dépendances entre les points de synchronisation et leur exécution.

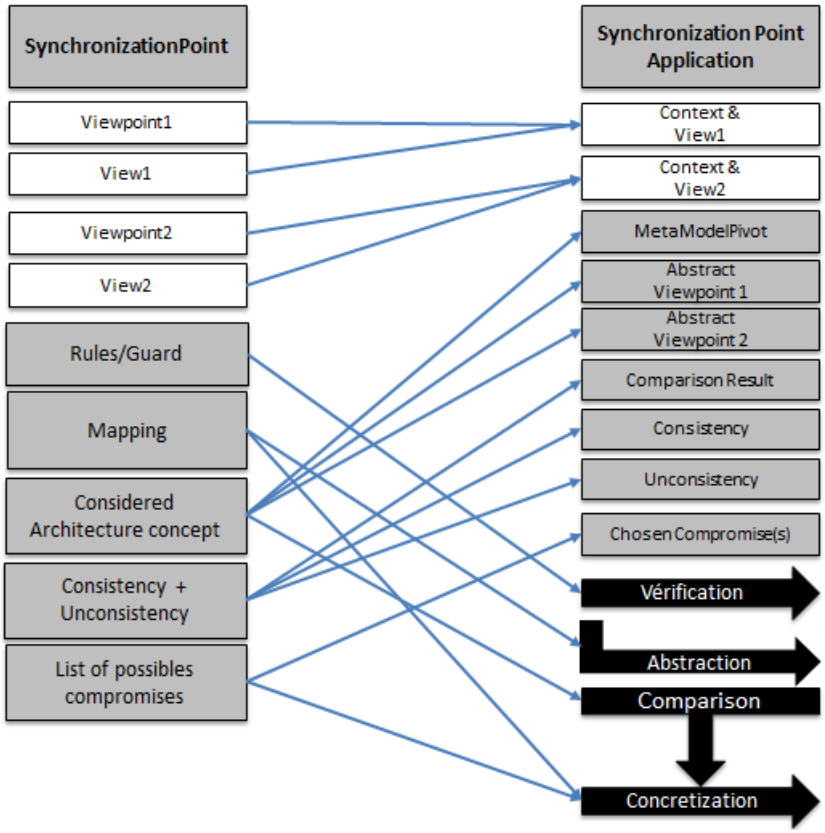


Figure 40 Dépendances des concepts de point de synchronisation et de l'application de la synchronisation

6.3. TRANSFORMATION DE MODELES

6.3.1. INTRODUCTION

La transformation de modèles *: c'est une des techniques clés de l'ingénierie des modèles. Elle consiste à prendre en entrée des modèles (source) et à fournir en sortie des modèles (cibles). Généralement un seul modèle source est utilisé et un seul modèle cible est fourni.*

Il existe plusieurs familles de transformation de modèles [112], [113]:

- Transformation endogène : Les modèles source et cible sont conformes au même métamodèle ;
- Transformation exogène : Les modèles source et cible sont conformes à des métamodèles différents ;
- Transformation verticale : Les modèles source et cible n'ont pas le même niveau d'abstraction ;
- Transformation horizontale : Les modèles source et cible ont le même niveau d'abstraction.

Ces familles sont illustrées dans la Figure 41.

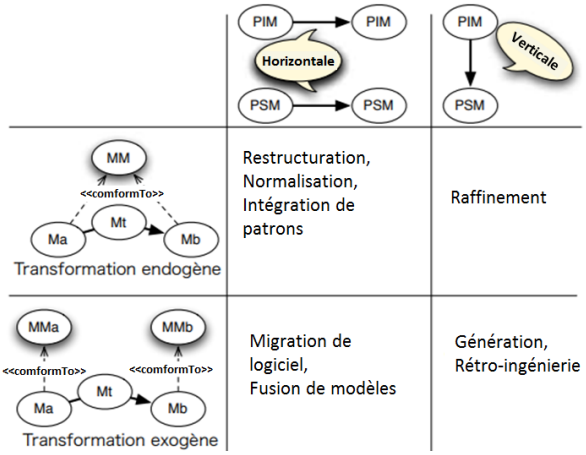


Figure 41 Types de transformations et leurs principales utilisations

Les transformations de modèles ne sont pas récentes et peuvent être chronologiquement classées en trois générations :

- Génération 1 : **Transformation séquentielle d'enregistrement.** Un script spécifie comment un fichier d'entrée est réécrit en un fichier de sortie (p. ex. des scripts Unix, AWK ou Perl) ;
- Génération 2 : **Parcours d'arbres**. Ces méthodes permettent le parcours d'un arbre d'entrée au cours duquel sont générés les fragments de l'arbre de sortie ;
- Génération 3 : **Transformation de graphes**. Avec ces méthodes, un modèle en entrée (graphe orienté étiqueté) est transformé en un modèle en sortie. Ces approches considèrent l'opération de transformation dans un troisième modèle *Mt* défini par un métamodèle *MMt* conforme aux métamodèles *MMa* et *MMb* des modèles source *Ma* et cible *Mb :*

$$Mb = f(MMa, MMb, Mt, Ma)$$

La dernière génération a donné lieu à d'importants travaux de recherche. Trois grandes techniques [113] de transformation en résultent :

- **Approche déclarative** : Elle recherche des patterns (d'éléments et de leurs relations) dans le modèle source. Chaque pattern trouvé est remplacé dans le modèle cible par une autre structure d'élément. Techniques : Triple Graph Grammars [114], EMF-IncQuery [115], [116], Acceleo 3[37], QVT-Relation [117].
- **Approche impérative** [118] : Elle parcourt le modèle source dans un certain ordre et lors de ce parcours le modèle cible est généré. Exemple : QVT-Operational [117].
- **Approche hybride** : Elle est à la fois déclarative et impérative. ATL [119], QVT [117].

Les transformations de modèles fournissent un moyen standard de transférer le contenu d'un modèle source vers un modèle cible. Cependant une fois appliquées, les modèles coexistent et évoluent indépendamment.

6.3.2. TECHNIQUE D'ABSTRACTION

L'exécution des abstractions de modèle parcourt la structure des modèles source pour la reconstruire à un plus haut niveau d'abstraction. C'est l'approche impérative [117] qui est la plus appropriée pour ce type d'activité. L'approche hybride peut aussi convenir mais elle n'a pas été testée dans cette thèse.

Trois étapes automatisables ont été définies pour permettre de définir les abstractions de modèle :

1. Valider la définition des modèles et métamodèles ;
2. Construction des fonctions de transformation ;
3. Organisation des fonctions de transformation.

La « validation des définitions des modèles et métamodèles » doit vérifier si la définition des contextes des disciplines d'ingénierie, notamment les vues et les points de vue respectent la configuration de la Figure 42. Les modèles (ou vues) doivent être conformes aux métamodèles. Les métamodèles doivent être conformes au méta-métamodèle. Le langage de transformation dépendra de la technologie choisie par l'outil de synchronisation.

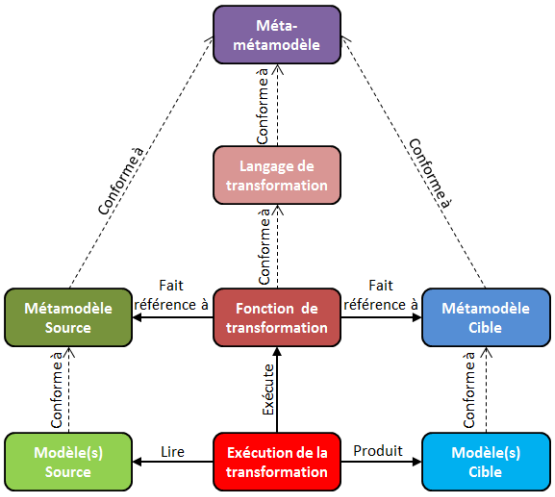


Figure 42 Métamodèle de transformation impérative

La « construction des fonctions de transformation » permet de définir toutes les opérations élémentaires de parcours des éléments sources pour construire les éléments cibles.

[37] [web37] Frederic Madiot (juin 2017). "Acceleo Tool", sur le site Eclipse. Consulté le 19 août 2017. https://wiki.eclipse.org/Acceleo

Une fonction de transformation *est une fonction F qui prend des éléments sources (X) particuliers et qui construit un ou plusieurs élément(s) cible(s) équivalent(s) (X') à partir des métamodèles. Elle définit également des conditions et des relations de traçabilité entre ces éléments.*

$$F(X, Condition) = (X', Relation\ de\ traçabilité)$$

avec X un élément du métamodèle source et X' un élément du métamodèle cible

Une représentation graphique est proposée pour définir chaque fonction de transformation. Elle s'inspire des travaux d'Eugène Syriani sur l'outil AtomPM [120]. La Figure 43 montre un exemple de fonction de transformation d'un block fonctionnel SysML en un hypothétique métamodèle de fonction.

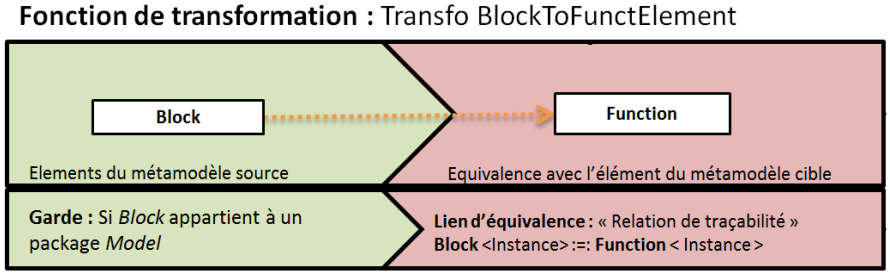


Figure 43 Représentation graphique d'une fonction de transformation

L'organisation des fonctions de transformation, en un processus, se construit selon le parcours du modèle source. La démarche top-down est la plus pertinente pour parcourir la structure du modèle source. La Figure 44 représente trois fonctions de transformation ordonnancée. Elle s'inspire des travaux d'Eugène Syriani sur l'outil AtomPM [120] et du langage BPMN [27].

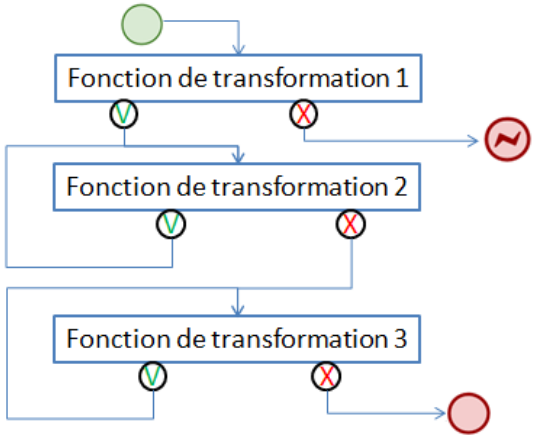


Figure 44 Ordonnancement des fonctions de transformation

6.3.3. TECHNIQUE DE CONCRETISATION

Pour réaliser des concrétisations, la même technique que l'abstraction est adoptée mais avec plusieurs modèles sources issus des compromis choisis et de la vue abstraite.

Les 3 mêmes étapes automatisables sont menées :

1. Valider la définition des modèles et métamodèles (identique à l'abstraction des modèles) ;
2. Construction des fonctions de transformation ;
3. Organisation des fonctions de transformation (identique à l'abstraction des modèles).

La « construction des fonctions de transformation » s'effectue de la même manière que pour l'abstraction de modèles à l'exception du fait qu'il y a plusieurs modèles sources : celui de la vue abstraite et celui des compromis. Les gardes peuvent être définies pour chaque modèle source ou pour tous les modèles sources. La Figure 45 présente un exemple de fonction de transformation spécifique à la concrétisation.

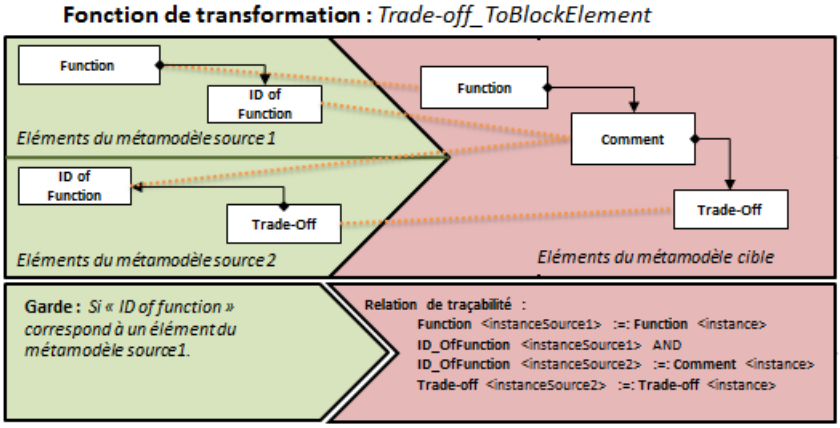


Figure 45 Exemple de fonction de transformation pour la concrétisation de modèle

6.4. COMPARAISON DE MODELES

L'exécution d'une comparaison de modèles parcourt la structure des modèles résultant de l'abstraction. C'est également une approche impérative.

Deux étapes ont été définies pour permettre de définir les comparaisons de modèles :

1. Valider la définition des modèles et métamodèles ;
2. Construction de l'algorithme de comparaison selon les concepts pivots d'architecture ;

La « validation des définitions des modèles et métamodèles » vérifie les modèles issus de l'abstraction. Ces modèles sont définis par le métamodèle pivot. La comparaison construit un modèle de cohérence qui contiendra des relations de cohérence et d'incohérence. Ce modèle de cohérence résulte de la saisie des relations tracées par les ingénieurs et les architectes. Les modèles et les métamodèles doivent respecter la configuration de la Figure 46.

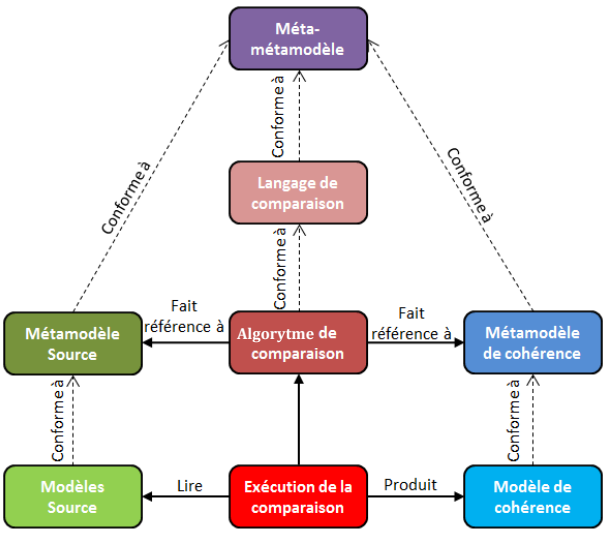


Figure 46 Métamodèle de comparaison de modèles

En fonction du métamodèle source, et donc des concepts pivots d'architecture un algorithme de comparaison est exécuté pour comparer les différents types d'éléments, de relations et propriétés.

Les algorithmes accompagnent les ingénieurs dans la mise en cohérence et construisent les relations de cohérence correspondantes.

7. TRAÇABILITE ET HISTOIRES DES MODELES

La traçabilité [121] a pour objectif de mémoriser les résultats des applications d'interactions multidisciplinaires effectuées. Elle trace les exécutions d'abstractions, de comparaisons (semi-automatiques) et de concrétisations au travers du déroulement successif des itérations et des points de synchronisation.

7.1. INTRODUCTION ET CONCEPTS DE LA TRAÇABILITE

La traçabilité selon l'ISO 8402 [122] et la norme NF X 50 120 [110] est l'aptitude à retrouver l'historique, l'utilisation ou la localisation d'un article ou d'une activité, ou d'activités semblables, au moyen d'une identification enregistrée. Aujourd'hui, elle est couverte par l'ISO 9000 [106], 9001 [123] et 9002 [124], c'est « l'aptitude à retrouver l'historique, la mise en œuvre ou l'emplacement de ce qui est examiné ».

Les technologies évoluant pour mieux répondre aux attentes, nous proposons la définition suivante :

Traçabilité : *capacité à suivre l'historique, l'utilisation, ou la localisation d'un flux d'informations au moyen d'une identification enregistrée. La traçabilité est représentée par des traces.*

En ingénierie des modèles, les principes de la traçabilité sont :

- Identifier des produits, des objets, des éléments, des modèles, des points de vue ;
- Enregistrer tous les liens successifs dans la chaîne ;
- Enregistrer les données concernant la traçabilité tout au long du cycle de vie des modèles ;
- Communiquer aux partenaires descendants l'information nécessaire et suffisante pour assurer la traçabilité du produit (ou du système).

La traçabilité [125] est nécessaire pour :

- La maîtrise de la qualité. Elle permet de retrouver la cause d'un écart. De plus, elle garantit la véracité des informations contenues dans les modèles.
- La logistique des modèles. La localisation, à chaque instant, d'éléments/entités des modèles permet d'optimiser la maîtrise des modèles et de leurs contenus. Elle permet aussi de suivre en temps réel ses évolutions, les jalons et les livrables.
- Le respect de la réglementation.

La gestion unitaire des éléments est la clé d'une traçabilité efficace. Elle permet un suivi fin.

Dans le contexte de cette thèse, la traçabilité exige que les éléments des modèles soient identifiables. Elle doit pouvoir référencer tous les contenus des modèles.

Une trace [126] contient un ensemble de liens de traçabilité traduisant des correspondances entre des éléments de modèles ou de vues employés par les différentes fonctions de synchronisation.

Trace : *ensemble de liens de traçabilité concernant une même activité (ici, les activités sont l'abstraction, la comparaison et la concrétisation)*

Lien de traçabilité : *une relation d'un ensemble d'éléments sources à un ensemble d'éléments cibles (ou de destination).*

La Figure 47 représente un modèle de lien de traçabilité entre des éléments.

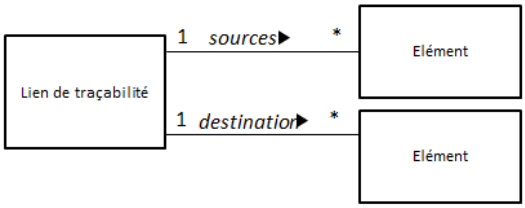


Figure 47 Modèle de Lien de traçabilité

La Figure 48 présente un modèle général de trace. On fait l'hypothèse que les traces peuvent faire le lien entre des éléments des modèles hétérogènes (traçabilité horizontale). C'est une différence majeure par rapport aux activités de traçabilité classiques qui retracent des éléments entre plusieurs versions ou plusieurs niveaux de raffinement d'un même modèle (traçabilité verticale).

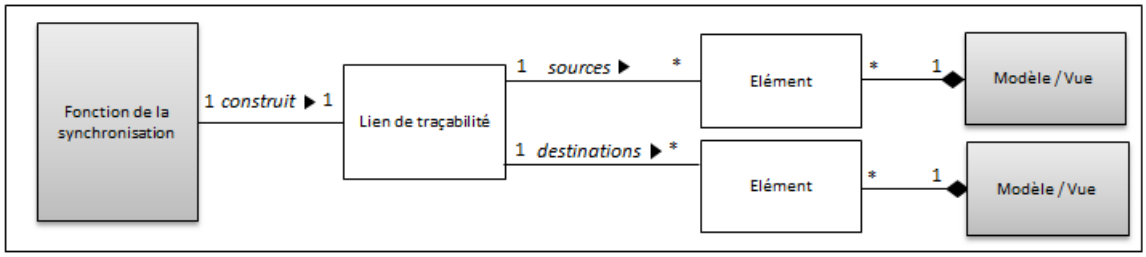


Figure 48 Modèle de définition d'une trace

Plusieurs catégories de traces sont définies pour capitaliser les activités réalisées durant la synchronisation de modèles :

- Traçabilité de synchronisation (Global) ;
- Traçabilité d'un point de synchronisation ;
- Traçabilité d'une itération ;
- Traçabilité d'abstraction ;
- Traçabilité de la comparaison ;
- Traçabilité de la concrétisation.

Ces catégories vont être expliquées dans la section suivante.

7.2. MODELISATION DES CONCEPTS

La traçabilité de synchronisation (Global) capitalise l'ordonnancement des points de synchronisation et les traçabilités des points de synchronisation. C'est la trace de plus haut

niveau, à partir de laquelle, il est possible d'obtenir toutes les autres traces. La Figure 49 représente le métamodèle de la traçabilité de synchronisation (Global).

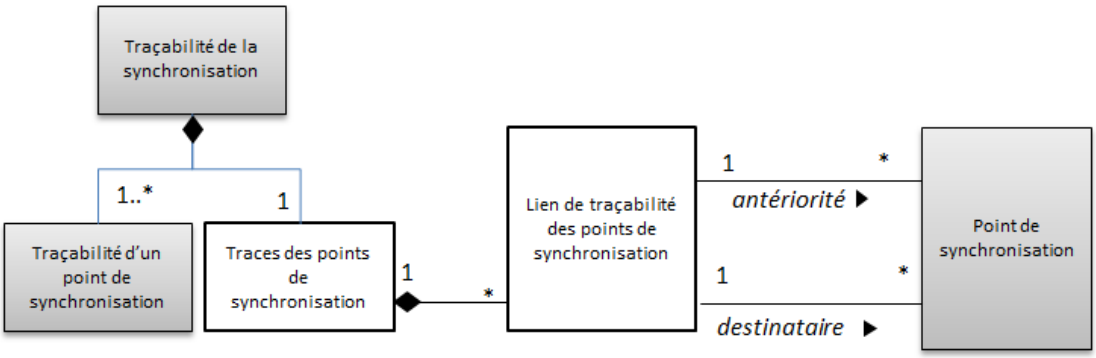


Figure 49 Métamodèle de la traçabilité de la synchronisation

La traçabilité d'un point de synchronisation capitalise toutes les traçabilités des itérations effectuées et l'ordonnancement des itérations exécutées. Les exécutions des itérations étant linéaires, l'ordonnancement peut être décrit sous la forme d'une liste ordonnée. La Figure 50 représente le métamodèle de la traçabilité d'un point de synchronisation.

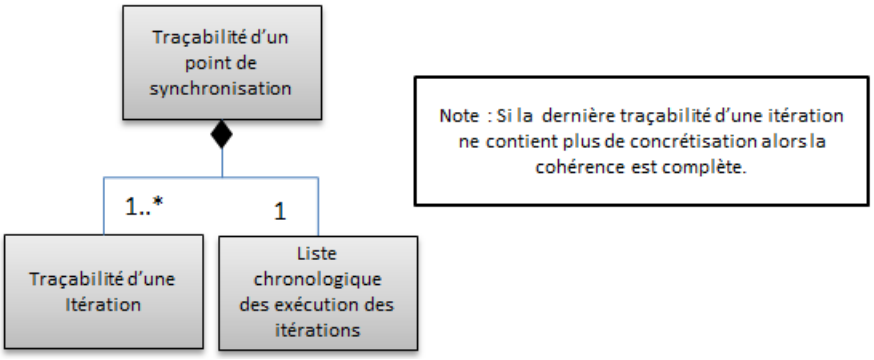


Figure 50 Métamodèle de la traçabilité d'un point de synchronisation

La traçabilité d'une itération réunit l'ensemble des traces produites par les abstractions des vues de chaque contexte, la comparaison et les concrétisations éventuelles vers les vues d'origines.

La traçabilité d'une itération, représentée dans la Figure 51, capitalise :

- les traces d'abstraction d'une vue du contexte A, à partir du métamodèle de A vers des concepts pivots d'architecture et d'un mapping.
- les traces d'abstraction d'une vue du contexte B, à partir du métamodèle de B vers des concepts pivots d'architecture et d'un mapping.
- les traces de comparaison, à partir du(des) concept(s) pivot(s) d'architecture et des saisies effectuées.
- les traces de la concrétisation vers la vue du contexte A, à partir de la liste de compromis du contexte B, du(des) concept(s) pivot(s) d'architecture et du métamodèle du contexte A.
- Les traces de la concrétisation vers la vue du contexte B, à partir de la liste de compromis du contexte A, du(des) concept(s) pivot(s) d'architecture et du métamodèle du contexte B.

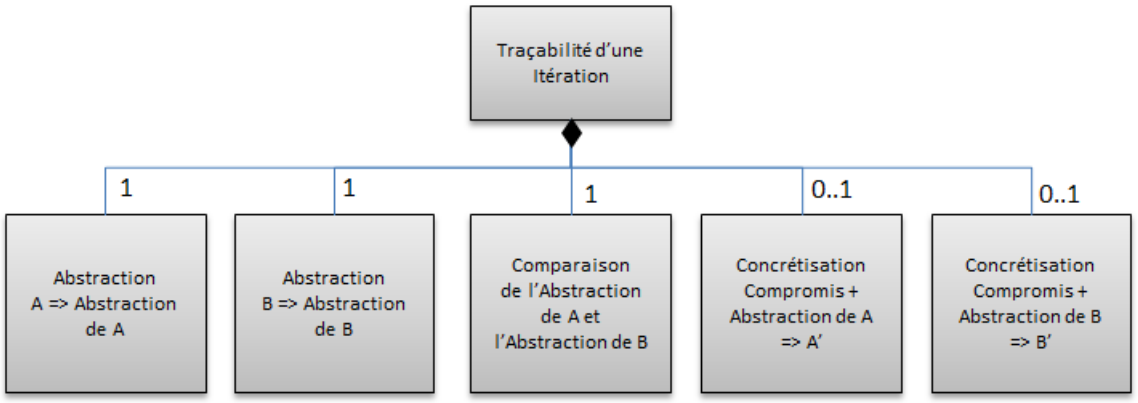


Figure 51 Métamodèle de la traçabilité d'une itération

Les traces d'abstraction des vues des contextes A et B sont définis de la même façon. Elles contiennent un ensemble de liens de traçabilité entre des éléments sources et des éléments cibles (« destination »). Ces éléments sont eux-mêmes contenus dans des modèles (vues) sources et un modèle abstrait. Le métamodèle de trace d'abstraction est présenté dans la Figure 52.

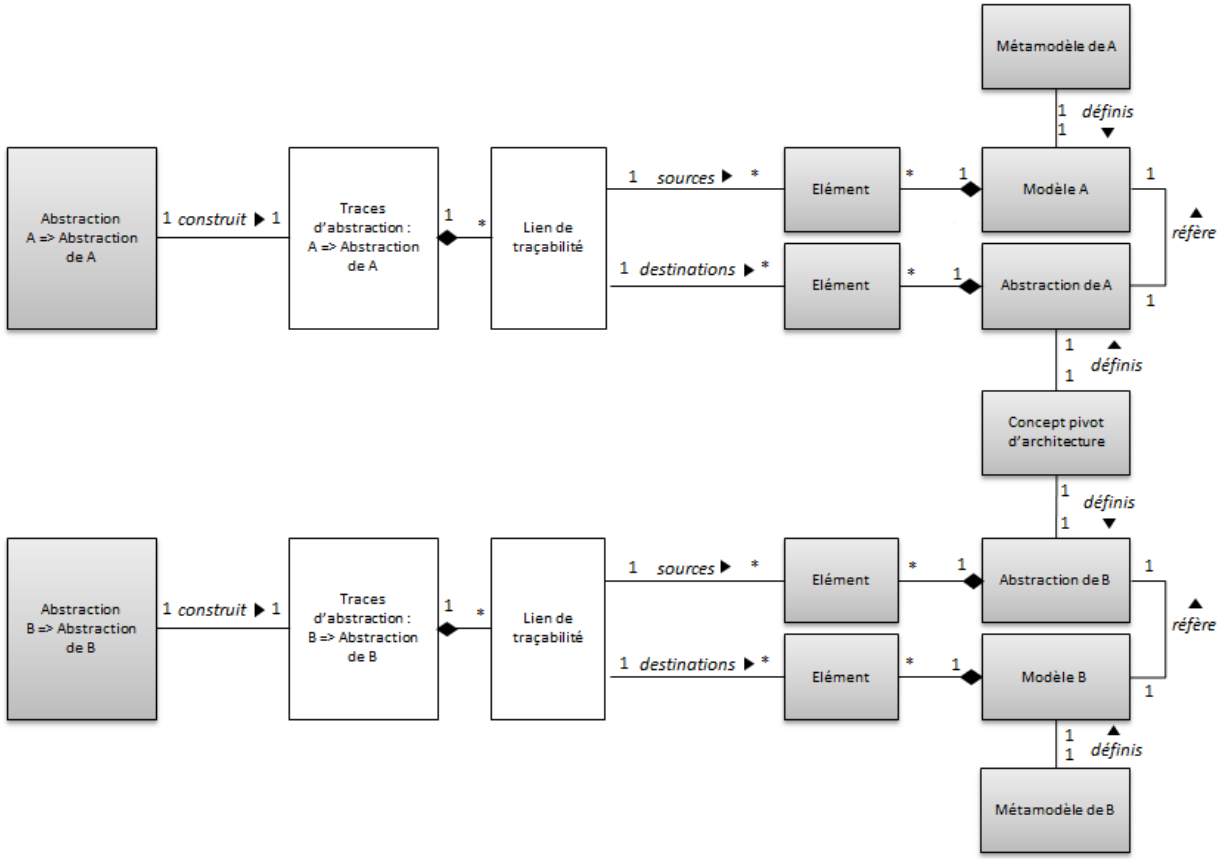


Figure 52 Métamodèle de la traçabilité des abstractions

Les traces de comparaison contiennent un ensemble de liens de traçabilité, qui peuvent être de deux types : les relations de cohérence et les incohérences. Ils sont construits à partir des saisies effectuées par les ingénieurs et architectes durant la comparaison.

Un lien de traçabilité relie les éléments des abstractions de chaque contexte. Si c'est une relation de cohérence, il peut également faire référence à une incohérence constatée dans une itération précédente. Le métamodèle de traces de comparaisons est présenté dans la Figure 53.

Un lien de traçabilité, de type incohérence capitalise en plus :

- une référence à un lien d'incohérence constatée dans une itération précédente ;
- une justification/argumentation saisie par des ingénieurs ou les architectes ;
- une caractérisation de la nature de l'incohérence.

La nature de l'incohérence est une manière de caractériser des types d'incohérence, e.g. un manque de concept dans une vue, un besoin de raffinement ou d'abstraction d'une vue, un besoin d'une réorganisation des éléments, etc.

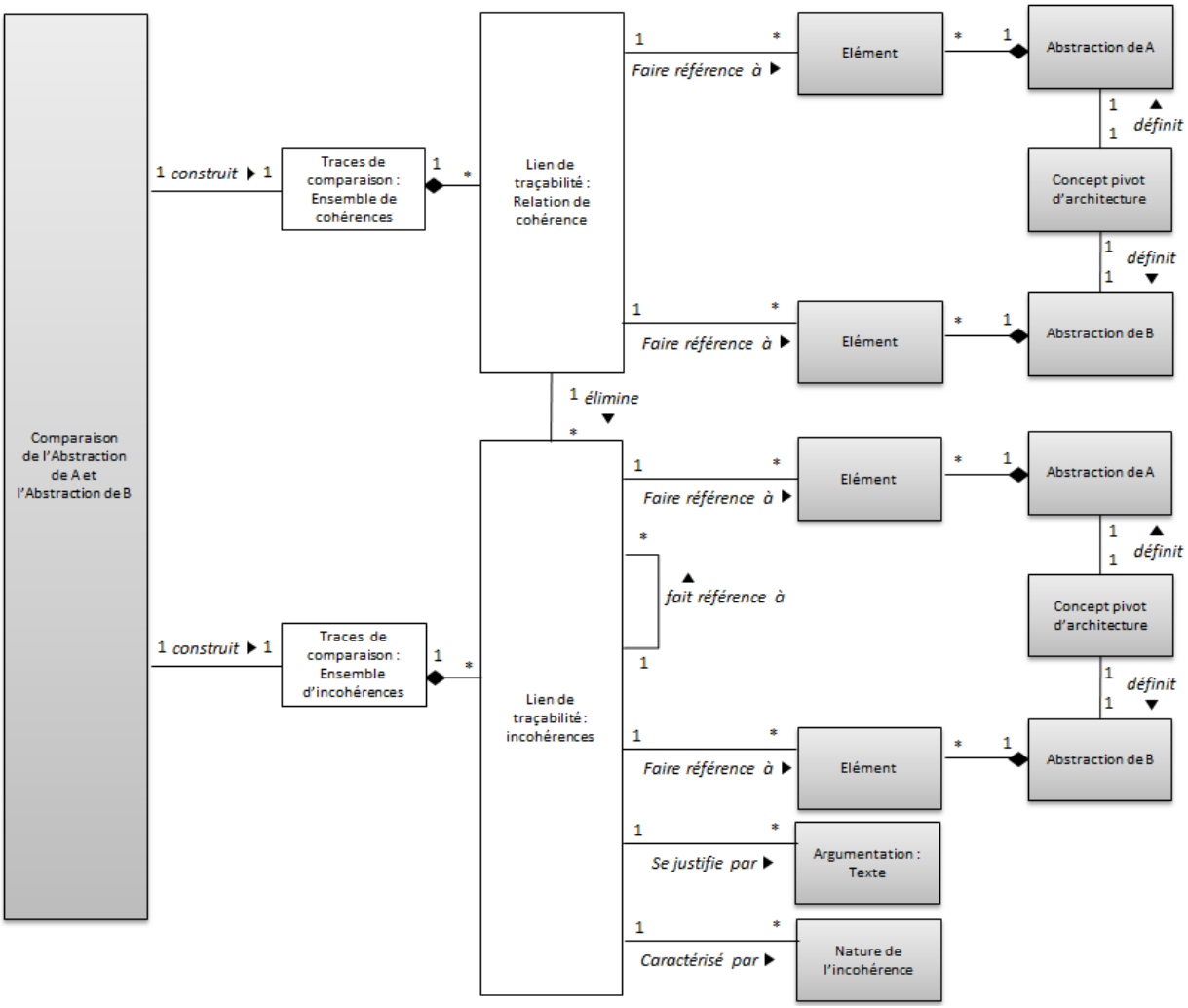


Figure 53 Métamodèle de la traçabilité de la comparaison

Les traces de concrétisation contiennent un ensemble de liens de traçabilité entre des éléments sources (abstraction des vues) et des éléments destinations (les vues) ainsi qu'un compromis. Une trace de concrétisation est issue d'une incohérence identifiée durant la comparaison. Le métamodèle de trace de concrétisation est présenté dans la Figure 54.

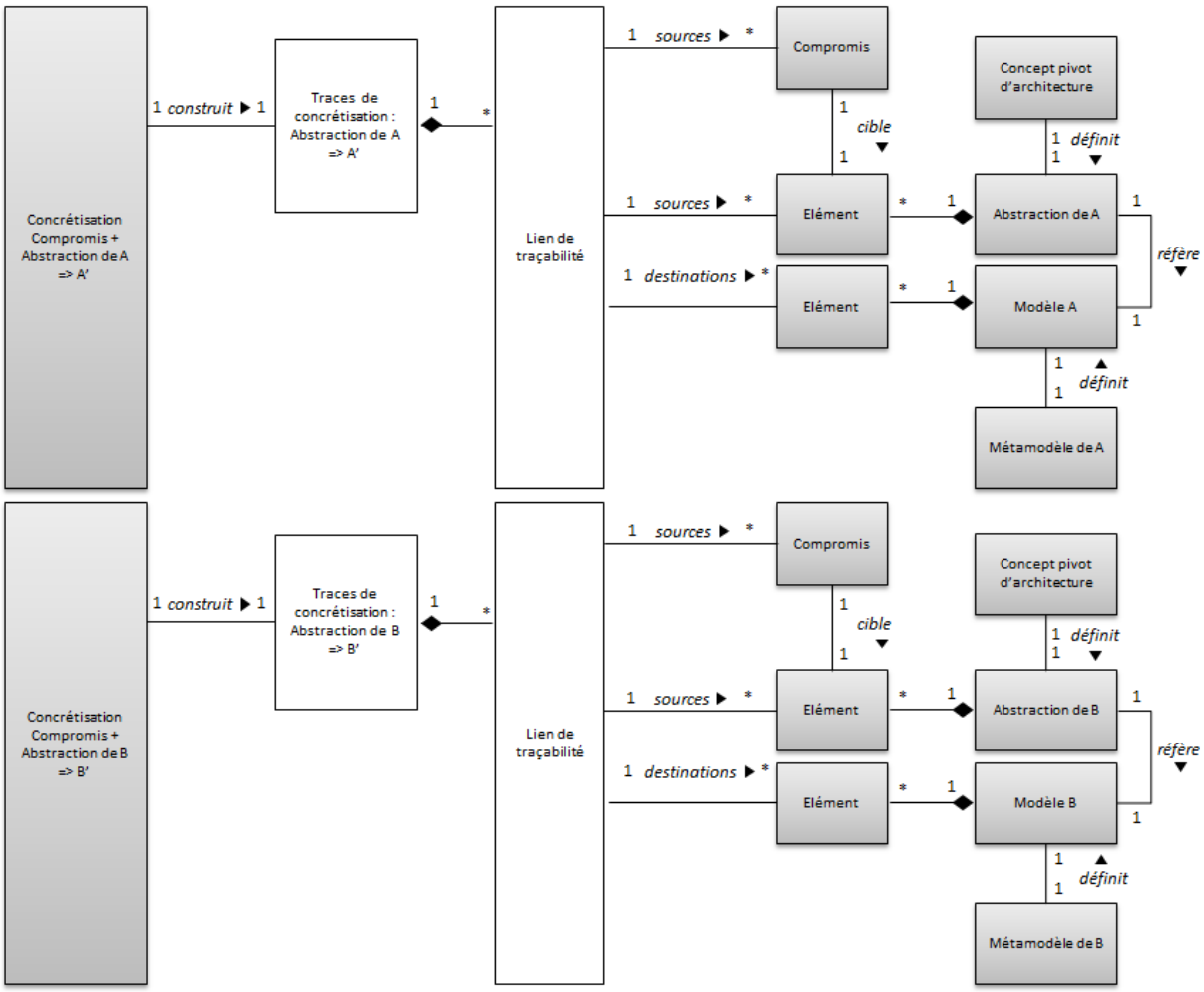


Figure 54 Métamodèle de la traçabilité des concrétisations

8. METHODOLOGIE DE SYNCHRONISATION DE MODELES

Pour permettre l'intégration de toutes les étapes définies dans ce chapitre, des concepts méthodologiques sont proposés. Ils permettront de définir une méthodologie qui peut se spécialiser au contexte de l'entreprise. Des concepts additionnels sont présentés pour initier la démarche du sponsor et du responsable de synchronisation. Ces derniers pourront plus facilement spécifier le besoin collaboratif des disciplines au niveau décisionnel.

Les concepts utiles existent déjà au travers de formalismes de modélisation ou de cadres d'architecture d'entreprise. Les propositions qui suivent s'inspirent des langages BPMN [27], UML [26], et de TOGAF 9.1 [103].

8.1. CONCEPTS METHODOLOGIQUES

***Méthodologie** : Branche de la logique étudiant les méthodes des différentes sciences [127]. C'est un processus générique, qui peut être décliné sous des formes plus spécifiques en un ensemble d'activités séquentiellement organisées.*

Le concept d'« activité » est défini dans la section Chapitre II2.2.

Livrable : *Résultats/Documents attendus après la réalisation d'une activité. Il contient les résultats de sortie de l'activité dans son contexte.*

Objectif : *Raison d'usage d'une activité, d'un processus ou d'une méthodologie. L'objectif doit être satisfait par les livrables produits.*

8.2. PRINCIPES DE SYNCHRONISATION

Plusieurs concepts sont introduits, pour que le responsable de synchronisation, accompagné par le sponsor, motive la mise en œuvre d'une démarche collaborative durant l'étape de lancement. Ils permettront de définir les disciplines d'ingénierie concernées, la vision opérationnelle du projet, le fonctionnement actuel des disciplines et le fonctionnement ciblé.

Les concepts de « partie-prenante », « disciplines d'ingénierie », « préoccupation », « système étudié », (cf. Chapitre II2.2) sont repris dans cette section.

Motivation du sponsor *: l'ensemble des facteurs déterminant le choix du sponsor à sélectionner une discipline d'ingénierie.*

Objectif du sponsor *: raison de la sélection des disciplines d'ingénierie par le sponsor pour la mise en œuvre d'une synchronisation.*

Mission *: les fonctions ou activités remplies par une discipline d'ingénierie sur un système.*

Usecase, *selon UML [26] : élément permettant de capturer les exigences des systèmes de manière informelle. Il décrit ce que les systèmes sont supposés faire.*

Les relations de structuration présentes dans les diagrammes de cas d'utilisation UML [26] sont également utilisées tel que : l'association, la relation « Extend » et la relation « Include ».

En complément, une déclinaison de « principes », dans TOGAF 9.1 [103], est proposée pour distinguer les intentions intrinsèques à la synchronisation de modèles et celles liées à l'entreprise.

Principe, *selon TOGAF 9.1 [103]:* *une déclaration qualitative d'une intention qui doit être satisfaite par la solution implémentée. Elle est accompagnée d'au moins une justification et une mesure d'importance.*

Principe Généraux : *un principe que l'approche satisfera indépendamment de l'entreprise ou du contexte.*

Principe Spécifique *: un principe spécifique au contexte et/ou à la volonté des parties-prenantes de l'organisme mettant en place cette démarche.*

8.3. MODELISATION D'UN METHODOLOGIE DE SYNCHRONISATION

Un formalisme de méthodologie est proposé dans la Figure 55. Il rassemble les concepts présentés dans la section « Concepts méthodologiques ».

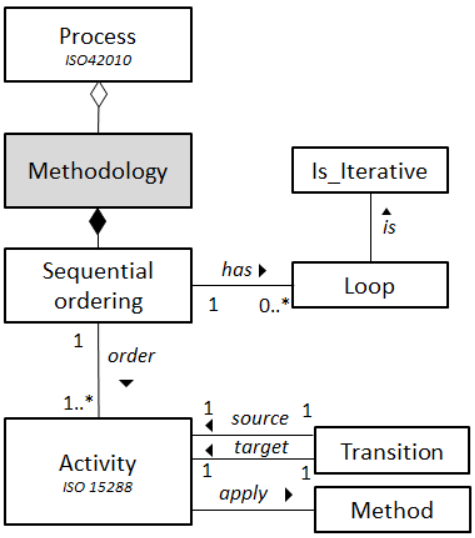


Figure 55 Métamodèle de la Méthodologie

Ce métamodèle sera utilisé dans le chapitre suivant pour définir une méthodologie applicable en entreprise.

8.4. MODELISATION DES PRINCIPES DE SYNCHRONISATION

Trois formalismes complémentaires sont proposés pour :

- Associer les parties prenantes, dans l'environnement du projet, à l'étape de lancement ;
- Décliner les principes généraux et spécifiques du projet ;
- Définir l'environnement opérationnel du projet et de ses interactions.

Ils utilisent et rassemblent les concepts de la section « Principes d'interactions ».

Le premier métamodèle, Figure 56, associe les différentes parties prenantes et les motivations du sponsor dans le choix des disciplines d'ingénierie. Il est utilisé par le sponsor et le responsable de synchronisation.

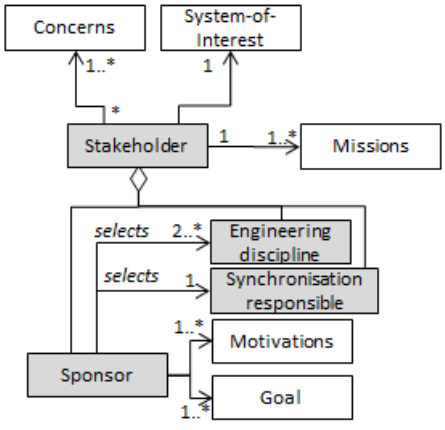


Figure 56 Métamodèle des parties prenantes

Le deuxième métamodèle, Figure 57, regroupe les concepts de principe et ses déclinaisons. C'est un premier pas vers la spécification des exigences d'un projet collaboratif.

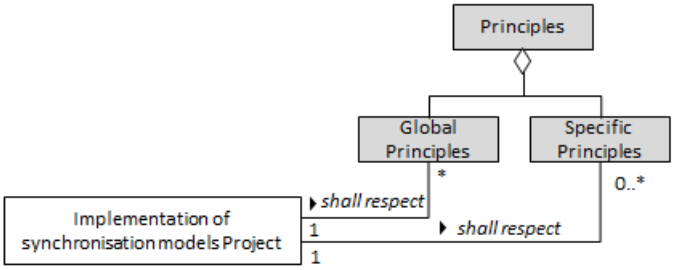


Figure 57 Métamodèle des principes

Le troisième métamodèle, Figure 58, est celui du diagramme de cas d'utilisation d'UML [26]. Il peut être utilisé pour définir l'environnement du projet de synchronisation dans une entreprise. La représentation graphique associée, proposée par l'OMG, n'est pas un prérequis. Ce sont surtout les concepts associés qui sont intéressants. D'autres diagrammes et formalismes existent, le diagramme UML [26] correspond à un candidat acceptable.

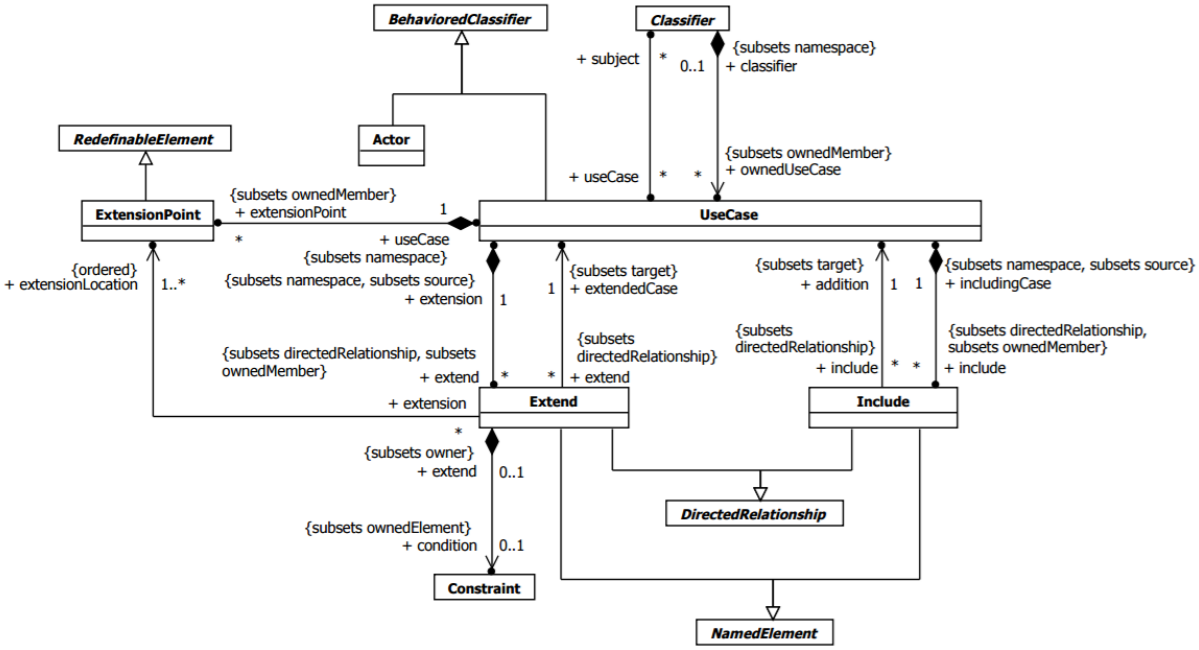


Figure 58 Métamodèle UML du diagramme cas d'utilisation

Chapitre III PROPOSITION D'UNE METHODOLOGIE DE SYNCHRONISATION DE MODELES

Ce chapitre présente une proposition de méthodologie d'interactions multidisciplinaires. Elle met en œuvre des synchronisations de modèles entre les vues des disciplines d'ingénierie. Elle est applicable dès lors qu'un dirigeant (sponsor) souhaite organiser des interactions entre des disciplines d'ingénierie. Elle peut notamment s'appliquer entre l'architecture système et la sureté de fonctionnement, c'est ce qui sera présenté au Chapitre V.

La méthodologie tient compte des standards (ceux qui sont spécifiques aux disciplines d'ingénieries, ceux qui sont liés au cycle de développement [35], [72]), de l'état de l'art scientifique (cf. Chapitre I), du cadre conceptuel (cf. Chapitre II) et des travaux du domaine de l'architecture d'entreprise (notamment TOGAF [103]).

1. INTRODUCTION GENERALE

1.1. OBJECTIF

L'introduction générale présente les grandes étapes de la méthodologie. La structure de la méthodologie s'inspire des cadres d'architecture d'entreprise présentés dans la section 1.2, tel que TOGAF 9.1 [103].

La méthodologie a pour objectif d'établir une mise en cohérence entre les modèles tout en étant un support aux interactions multidisciplinaires.

1.2. ARCHITECTURE D'ENTREPRISE

L'architecture d'entreprise permet de mettre en œuvre des solutions organisationnelles dans l'entreprise en tenant compte de l'état des pratiques et des facteurs d'impact (opérationnel, fonctionnel, technologique et d'infrastructure). Elle propose aussi des architectures intermédiaires pour la conduite du changement.

TOGAF [103] est une méthodologie et un cadre d'architecture d'entreprise éprouvé et utilisé par les principales organisations internationales pour améliorer l'efficacité de leurs entités. TOGAF accompagne les architectes d'entreprise avec des méthodes adaptatives.

Suivant les domaines d'application, les cadres d'architecture sont déclinés en :

- TOGAF 9.1 [103] proposé par the Open Group.
- DODAF [128] appliqué aux programmes d'armes et de systèmes d'information du *Departement of Defense* des Etats-Unis ;
- MODAF [129] appliqué aux systèmes, aux systèmes de systèmes et aux processus d'affaires du *Ministry of Defence* de Grande Bretagne ;
- NAF [130] proposé par l'OTAN. La version 3.1 est notamment appliquée en France par la direction générale de l'armement, DGA.

1.3. PROPOSITION METHODOLOGIQUE

Des réflexions ont permis d'explorer :

- Les moyens pour définir, configurer et mettre en œuvre des synchronisations de modèles ;
- Les applications possibles de la synchronisation de modèles hétérogènes ;
- L'exploitation des résultats issus d'applications de synchronisation et les moyens de fournir des histoires sur l'évolution des modèles durant le cycle de développement.

Ces axes d'étude ont permis de retenir une méthodologie générale. Il s'agit d'une approche itérative au niveau des activités de la méthodologie et éventuellement de ses sous-parties.

La méthodologie et ses activités sont adaptatives, i.e. elles évoluent selon le besoin et le contexte dans lequel elles s'appliquent. Elles guident et proposent des méthodes soulignant des points pertinents à traiter.

La méthodologie doit permettre des applications de synchronisation de modèles tout en respectant les contraintes suivantes :

- Maintenir la séparation des préoccupations, i.e. aucune interaction ne soit menée pendant que les ingénieurs et les architectes effectuent leurs études ou les analyses de leurs processus respectifs ;
- Contrôler les interactions et maîtriser les prises de décisions des ingénieurs lors de synchronisations de modèles ;
- Supporter un dialogue à propos de l'architecture du système entre les ingénieurs et les architectes ;
- Permettre la proposition d'évolutions de contenus des modèles ;
- Proposer des interactions bidirectionnelles qui n'imposent pas de niveau d'abstraction d'une discipline d'ingénierie à l'autre. Cependant, elle peut l'autoriser si la situation le demande.

Pour considérer tous ces aspects, 5 activités sont définies dans la méthodologie :

- Définition des principes d'interactions :
  - Elle permet d'identifier les disciplines d'ingénierie qui interviendront ;
  - Elle permet de définir les principes et la politique mise en avant pour interagir.
- Définition du contexte des disciplines d'ingénierie :
  - Elle détermine chaque contexte des disciplines d'ingénierie en termes de processus, d'activités, de méthodes et de points de vue ;
  - Elle identifie qualitativement les besoins de synchronisation.
- Configuration de synchronisation :
  - Pour chaque besoin, elle définit le ou les point(s) de synchronisation(s) correspondants ;
  - Elle assure le respect des mécanismes de synchronisation (abstraction, comparaison, concrétisation).
- Application de la synchronisation :
  - Elle met en œuvre la synchronisation de modèles ;
  - Elle capture les traces issues de ces activités.
- Suivi de la cohérence et évolution de la synchronisation :
  - Elle analyse des traces pour faire évoluer le processus de synchronisation.

Chacune des activités sera détaillée dans ce chapitre de la manière suivante :

- Objectif de l'activité ;
- Proposition de méthodes qui exploitent le cadre conceptuel de synchronisation ;
- Exemples simples d'application.

La méthodologie de synchronisation est présentée dans la Figure 59 et la Table 11. Elle réutilise les concepts et les modèles présentés dans la section « Méthodologie de synchronisation de modèles » (cf. Chapitre II8).

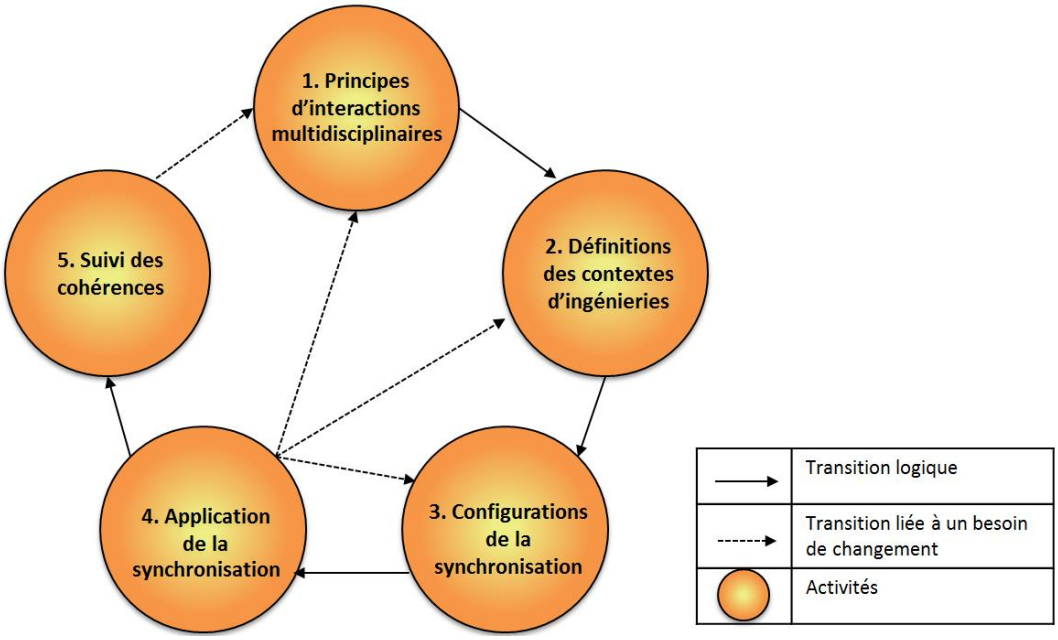


Figure 59 Méthodologie de synchronisation de modèles

Table 11 Définition des buts et des livrables des activités de la méthodologie de synchronisation de modèles

ID	Activités	But	Livrables
1	Principes d'interactions multidisciplinaires	Définir les objectifs des interactions.	- Liste d'objectifs d'interactions - Sélection des disciplines d'ingénierie
2	Définitions des contextes des disciplines d'ingénierie	Caractériser l'environnement de synchronisation.	Pour chaque discipline d'ingénierie : - Processus / activités / méthodes / points de vue - Liste des points de vue - Liste des besoins de synchronisation
3	Configuration de la synchronisation	Caractériser le besoin de synchronisation pour mise en application.	Pour chaque besoin de synchronisation : - Définition des points de synchronisation ; - Ensemble de concepts pivots d'architectures ; - Ensemble de compromis par disciplines ; - Ensemble des mappings.
4	Application de la synchronisation	Mise en œuvre de la synchronisation dans le projet de conception de système.	Pour chaque point de synchronisation : - Suivi de l'histoire de la synchronisation ; - Listes des relations de cohérence et des incohérences antérieures ; - Liste des incohérences.
5	Suivi et évolution des cohérences	Analyse du retour d'expérience.	- Proposition d'évolution ou de modification ; - Proposition d'extension de la synchronisation ; - Proposition d'introduction d'une nouvelle discipline.

L'intérêt premier de cette méthodologie est de faire intervenir des disciplines définissant ou exploitant l'architecture d'un système, indépendamment du paradigme de structuration du langage ou métamodèle employé. En spécialisant et en adaptant la méthodologie, il pourrait être envisageable de mener des interactions sur d'autres concepts que l'architecture, e.g. la conformité des exigences ou la description comportementale des composants.

1.4. EXEMPLES D'APPLICATION

Dans cette section, les scénarios A à E présentent des situations dans lesquelles la méthodologie peut être appliquée. A l'inverse le scénario F présente un contre-exemple.

Scénario A- Un systémier en aéronautique souhaite monter une démarche d'ingénierie système permettant de proposer des solutions collaboratives durant la réalisation d'activités transverses aux disciplines d'ingénierie. Ces solutions doivent accompagner les discussions tout au long du cycle de développement de systèmes complexes. Il souhaite mener une politique collaborative entre ces disciplines de manière progressive. Etant donné l'impossibilité des projets de conception à s'arrêter ou à ralentir significativement, l'entreprise souhaite mener une conduite du changement progressive avec un objectif de synchronisation des disciplines à long terme. Elle tient toutefois à éviter la rupture et attend du responsable de synchronisation des solutions collaboratives de transition.

Scénario B- Le chef de département a constaté un conflit d'intérêt entre les études menées par deux équipes. Le jugement et les analyses sont biaisés par le rôle de l'équipe A trop fort vis à vis de l'équipe B. A impose un certain nombre de résultats sans évaluer la faisabilité ou la cohérence des travaux avec l'équipe B.

Scénario C- L'équipe SLI ne sait pas comment ses processus s'inscrivent dans le processus de conception et de suivi opérationnel des projets, du moins cela a souvent été informel et non explicite. Le chef de division SLI souhaite construire un processus pour définir clairement les missions et les interactions avec les autres disciplines. Il veut mettre en avant une approche collaborative entre ses équipes et les expertises voisines.

Scénario D- Une jeune entreprise souhaite étendre les applications des systèmes qu'elle développe. De nouvelles contraintes de certification doivent être prises en compte, aujourd'hui, une seule équipe joue le rôle de concepteur et d'évaluateur. Elle a besoin de séparer les préoccupations et veut proposer des interactions fortes entre ces deux futures équipes tout au long du cycle de développement pour un gain de temps d'échanges tout au long du projet.

Scénario E- Un systémier et un équipementier souhaitent faciliter leurs collaborations en menant des discussions communes entre le client et les fournisseurs de solutions.

Scénario F- La société X souhaite construire un processus global pour chacun de ses projets en imposant des niveaux d'abstraction aux disciplines d'ingénierie par une configuration maitre-esclave. Les modèles employés seront construits par un modèle unique encapsulant l'ensemble des disciplines et pouvant lier l'ensemble des concepts.

Il n'est pas recommandé d'appliquer la méthodologie sur le scenario F car elle ne supporte par les approches « modèle unique » ou « maitre-esclave ».

2. ACTIVITE DE DEFINITIONS DES PRINCIPES D'INTERACTIONS

2.1. OBJECTIF

La première activité de la méthodologie, « Principes d'interactions multidisciplinaires », est très générique et qualitative. Elle permet de définir les objectifs et les motivations des parties prenantes. Elle permet aussi d'évaluer l'adéquation entre l'approche proposée et la vision cible souhaitée par le(s) sponsor(s).

Les méthodes, proposées dans la section suivante, peuvent évoluer ou être complétées par d'autres méthodes selon les besoins et les contextes spécifiques de l'entreprise.

Cette activité utilise les concepts et les modèles présentés à l'étape « Méthodologie de synchronisation de modèles » (cf. Chapitre II8).

2.2. METHODES

Six étapes sont proposées pour répondre aux objectifs de l'activité :

- Choix des disciplines d'ingénierie ;
- Définition opérationnelle du projet de synchronisation ;
- Etat actuel des interactions et des problèmes observés ;
- Définition de l'état cible ;
    - Lire les principes généraux de la méthodologie et sélectionner parmi les optionnels ceux qui seront suivis ;
    - Définir, avec les parties prenantes, les principes spécifiques ;
- Evaluation de l'adéquation du besoin avec l'approche ;
- Etudes complémentaires.

L'activité produit les livrables suivants :

- La liste des disciplines d'ingénierie et les motivations du sponsor ;
- La description des rôles et missions des disciplines ;
- Le diagramme de cas d'utilisation ;
- Les définitions des cas d'utilisation ;
- La description de l'état des pratiques actuelles ;
- La liste des principes généraux retenus ;
- La liste des principes spécifiques ;
- L'évaluation de l'adéquation du besoin avec l'approche ;
- Les résultats d'études complémentaires.

2.2.1. CHOIX DES DISCIPLINES D'INGENIERIE

Le sponsor et le responsable de synchronisation doivent identifier les disciplines d'ingénierie ciblées.

Dès lors qu'il peut exister une relation entre les modèles employés entre plusieurs disciplines d'ingénierie, il est pertinent de s'interroger sur le rapport et les relations entre ces disciplines. Il est possible que la synchronisation ne soit pas toujours la solution la plus adaptée, e.g. si les disciplines n'appliquent pas de démarche model-based. Toutefois même dans ce cas, la méthodologie peut permettre un accompagnement manuel d'un processus de synchronisation adapté au contexte de l'entreprise.

Le choix des disciplines doit être motivé par le sponsor. Il doit formuler les raisons et les objectifs pour chaque discipline qu'il souhaite intégrer dans la démarche.

Le nombre de disciplines sélectionnées n'est pas limité par la méthodologie, cependant il est judicieux de commencer avec deux disciplines et d'intégrer progressivement d'autres disciplines lors d'une nouvelle itération si cela est possible.

Les disciplines d'ingénierie sélectionnées doivent être décrites par :

1. une définition ;
2. leurs missions ;
3. leurs préoccupations ;
4. leur rôle métier ;
5. le système étudié.

2.2.2. DEFINITION OPERATIONNELLE DU PROJET DE SYNCHRONISATION

Pour définir l'organisation et les rôles des parties prenantes, il est pertinent de définir les cas d'utilisation et leurs associations avec les parties prenantes. Le sujet qui nous intéresse ici est le projet de mise en œuvre d'interactions en entreprise. Au sein de celui-ci, on retrouve toutes les activités de la méthodologie.

Cette méthode peut être réalisée avec des représentations graphiques telles que le diagramme de cas d'utilisation d'UML [17]. Un individu ne peut pas incarner plusieurs parties prenantes ni plusieurs disciplines d'ingénierie.

Les parties prenantes doivent être une déclinaison des acteurs (cf. Chapitre II1.1) suivants :
- Le sponsor qui est le commanditaire et financeur du projet ;
- Le responsable de synchronisation, qui est le responsable du projet ;
- Les disciplines d'ingénierie soumises à la synchronisation.

Chaque individu ne peut pas incarner qu'une seule partie prenante.

2.2.3. ETAT ACTUEL DES INTERACTIONS ET LES PROBLEMES OBSERVES

L'évaluation des efforts requis pour réussir le projet peut se caractériser par l'écart entre l'état des pratiques actuelles des interactions entre les disciplines choisies et l'état des pratiques ciblées ou envisagées. Pour cela, il faut donc d'abord décrire l'état des pratiques actuelles.

L'application de la méthode décrit qualitativement le déroulement des interactions, des discussions et des interactions entre les disciplines, tel qu'il est mené actuellement dans l'entreprise. Cette description doit répondre aux questions suivantes :

1. Existe-t-il des interactions entre les disciplines d'ingénierie ? Si oui, sont-elles formalisées ?
2. Sous quelle forme se traduisent ces interactions dans les projets ? (Réunions d'avant-projet ou de coordination, échanges documentaires, réunions régulières, échanges de modèles, autres)
3. A quel(s) moment(s) les interactions sont-elles menées (entre les disciplines) dans les différentes étapes de conception ?
4. Est-ce que les interactions entre les disciplines se concrétisent de manière unidirectionnelle ou bidirectionnelle ? Autrement dit, est-ce qu'une discipline impose ses représentations aux autres ou essaient-elles de construire ensemble une adéquation entre leurs informations communes ? La question peut se décliner par typologie d'échanges constatés si nécessaire. Il est possible que certaines interactions soient menées de manières unidirectionnelles et d'autres bidirectionnelles.
5. Quels risques ou problèmes sont observés sur les interactions effectuées (difficultés de communication, emplois de vocabulaire et de concepts techniques hétérogènes, modélisation manuelle à partir d'autres modèles, risques d'erreurs, temps de compréhension et d'appréhension trop longs, maitrise de la complexité difficile, autres) ?

2.2.4. DEFINITION DE L'ETAT CIBLE

Pour les mêmes raisons que précédemment, il est nécessaire de définir l'état des pratiques ciblées. Pour ce faire, la méthode suivante propose de définir les principes généraux et spécifiques en 2 étapes successives.

1. *Lire les principes généraux de la méthodologie et sélectionner parmi les principes optionnels ceux qui seront appliqués.*

Les principes généraux sont détaillés ci-dessous, ils peuvent être raffinés si besoin :

- Global :
  - Suivre une démarche collaborative ;
  - Guider et faciliter la communication ;
  - Centrer la synchronisation sur les architectures du système ;
  - Garantir un niveau de cohérence ;
  - Faciliter la gestion de la complexité ;
  - Permettre des interactions unidirectionnelles ou bidirectionnelles du contenu des modèles **(optionnel)**.
- Processus :
  - Renforcer les relations entre domaines d'ingénierie durant les étapes amont du projet **(optionnel)** ;
  - Favoriser l'investissement en étape amont, pour faciliter les interactions entre domaines d'expertise **(optionnel)**.
- Modèles :
  - Mettre en cohérence des modèles hétérogènes ;
  - Permettre l'interopérabilité des modèles ;
  - Origines des modèles connues ou inconnues, communes ou non **(optionnel)**.
- Décision :
  - Ouvrir/Renforcer/Imposer le dialogue entre les disciplines d'ingénierie choisies ;
  - Limiter les jugements/biais dans les études ;
  - Gérer/limiter/contrôler la prise de décision des disciplines d'ingénierie **(optionnel)**.
- Résultat :
  - Apporter une justification de la mise en cohérence des études.

2. *Définir avec les parties prenantes et les principes spécifiques.*

De la même manière que les principes généraux sont présentés, des principes spécifiques peuvent être définis. Ils correspondent aux besoins et aux contraintes des parties prenantes du projet.

Principes spécifiques :
- Global : …
- Processus : …
- Modèles : …
- Décision : …
- Résultat : …

Si les principes sont dépendants des activités des processus ou des pratiques d'une discipline, il est possible de décliner les principes selon d'autres critères que ceux présentés.

Enfin, il est judicieux de faire valider les principes (généraux et spécifiques) par le sponsor et les disciplines d'ingénierie concernées. De même, il est recommandé de vérifier que les principes ne sont pas contradictoires entre eux.

A la suite de ces étapes, le responsable de synchronisation détient un ensemble de principes qui constituent les objectifs opérationnels à mettre en œuvre.

2.2.5. EVALUATION DE L'ADEQUATION DU BESOIN AVEC L'APPROCHE

Cette étape a pour but d'évaluer l'adéquation du besoin avec l'approche. Les principes généraux sont déjà supportés par la méthodologie. Ce n'est peut-être pas le cas des principes spécifiques. Il est donc nécessaire de vérifier la possibilité de les prendre en compte.

Pour ce faire, la méthode propose d'allouer chaque principe spécifique avec la ou les activité(s) de la méthodologie qui permettront de le prendre en compte. Un tableau à deux dimensions avec en colonne la liste des principes spécifiques et en ligne les activités de la méthodologie peut être construit.

Si un principe spécifique n'est pas supporté par la méthodologie, il est recommandé de réitérer les différentes activités pour étendre le champ d'application.

2.2.6. ETUDES COMPLEMENTAIRES

Des études plus approfondies peuvent être menées afin d'enrichir l'activité « Principes d'interactions multidisciplinaires » selon les préoccupations de l'entreprise. Voici quelques exemples de méthodes complémentaires :

- Analyse budgétaire ;
- Analyse de faisabilité ;
- Analyse de risques projet ;
- Définitions des exigences.

2.3. EXEMPLES D'APPLICATION

La section présente plusieurs exemples simples d'applications des méthodes présentées.

Exemples d'application de la méthode 1 : Choix des disciplines d'ingénierie :

Pour le choix des disciplines d'ingénierie, le sponsor annonce l'objectif du projet et ses motivations concernant le choix des disciplines engagées dans le projet (cf. les scénarios d'exemple 1.4). Voici quelques exemples de choix de discipline d'ingénierie :

- Exemple 1 : Dans un processus de conception système, l'architecture système et l'analyse de la sûreté de fonctionnement ;
- Exemple 2 : Dans un processus analyse de risques, l'analyse de la sûreté de fonctionnement et les études du soutien logistique intégré ;
- Exemple 3 : Dans un processus analyse de la sécurité, la sécurité intrinsèque et la sécurité règlementaire ;
- Exemple 4 : Dans un processus d'analyse de sûreté de fonctionnement, l'évaluation de la fiabilité et l'évaluation de la maintenabilité ;
- Exemple 5 : Dans un processus de conception d'architecture, l'analyse opérationnelle système et la conception d'architecture fonctionnelle ;
- Exemple 6 : Dans un processus de conception d'architecture, la conception d'architecture fonctionnelle et la conception d'architecture organique ;
- Exemple 7 : Dans un processus de conception système, l'ingénierie des exigences et de l'architecture système.

Le rôle et les missions des acteurs des disciplines choisies sont ensuite caractérisés. Voici deux exemples.

- Exemple 1 : l'architecte système est responsable de créer une solution d'architecture répondant aux besoins qui lui sont fournis. En plus, il coopère avec les différentes parties prenantes du projet et adapte ses représentations d'architecture selon les parties prenantes et leurs centres d'intérêts ;
- Exemple 2 : l'évaluateur de la sûreté de fonctionnement prévisionnelle est responsable d'apprécier les performances de sûreté de fonctionnement, i.e. l'aptitude d'un système à remplir une ou plusieurs fonctions requises dans des conditions données à un instant donné.

Exemple d'application de la méthode 2 : Définition opérationnelle du projet de synchronisation

Le responsable de synchronisation doit définir les cas d'utilisation dans le cadre de la mise en œuvre du projet. La Figure 60 présente le diagramme d'utilisation résultant du travail du responsable.

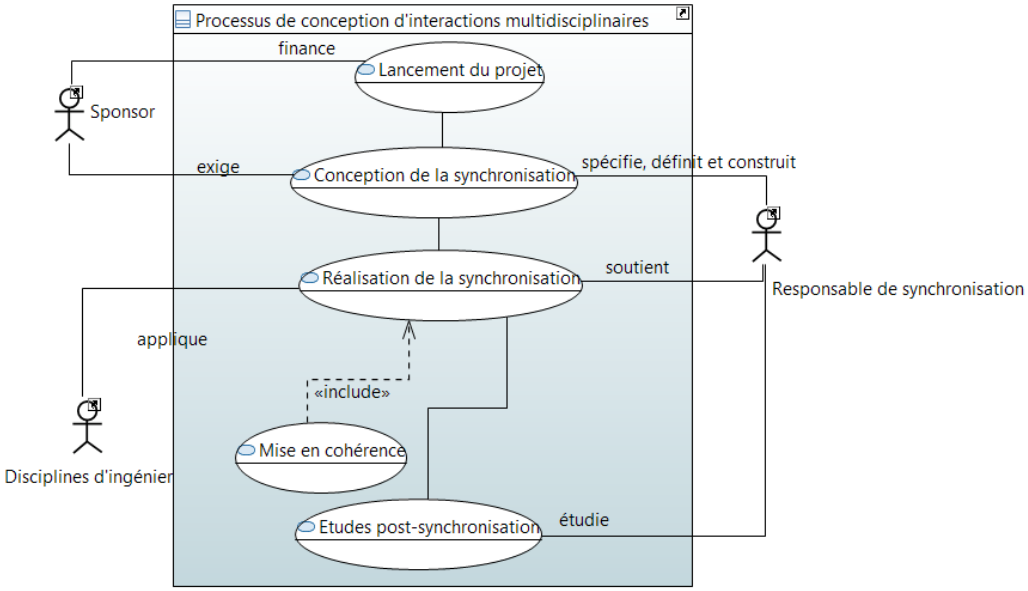


Figure 60 Diagramme de cas d'utilisation du processus de synchronisation de modèles

Exemple d'application de la méthode 3 : Etat actuel des interactions et des problèmes observés

Le responsable de synchronisation définit l'état actuel des interactions et des problèmes observés entre les disciplines d'ingénierie. Après avoir questionné les ingénieurs, le responsable de synchronisation fait une synthèse de la situation.

Les architectes témoignent (il s'agit de témoignages réels et anonymes, issus d'une enquête, cf. Annexe 1):

- *« Analyse de sûreté de fonctionnement non pertinente aboutissant à des exigences de safety inadéquates » ;*
- *« Lorsque l'interaction est inexistante, les études de sûreté de fonctionnement sont mal spécifiées (ex, sur-spécification du design, les études de études sûreté de fonctionnement n'ont alors plus aucun sens) » ;*
- *« Remise en question de la solution au moment de la vérification / exigences non-fonctionnelles dont la sûreté de fonctionnement obligeant une re-conception globale de la solution » ;*

- *« Re-design de composant, consommation de temps en réunion pour redéfinir les besoins tardivement dans le développement ».*

Les ingénieurs de sûreté de fonctionnement témoignent (il s'agit de témoignages réels issus d'une enquête, cf. Annexe 1):

- « Je me suis retrouvé dans un projet en étant en partie responsable des deux fonctions: architecture système <et> sureté de fonctionnement, au sein d'une équipe » ;
- « J'ai été amené à intervenir afin de proposer une nouvelle architecture sûre qui réponde aux exigences de fiabilité et de sûreté. L'architecture existante n'avait pas pris en compte ces aspects et n'était donc pas qualifiable ni certifiable ».

Exemple d'application de la méthode 4 : Définition de l'état cible

Le responsable de synchronisation doit ensuite définir l'état cible, c'est-à-dire l'objectif du projet. Il prend soin de bien saisir ce qu'impliquent les principes généraux. Il fait un choix sur les principes optionnels en fonction du besoin du sponsor. Il décide également d'ajouter deux principes spécifiques :

- Processus :
  - PS1 - Les interactions seront effectuées avant chaque jalon dans le processus pour garantir la cohérence des activités avant leur validation.
- Résultat :
  - PS2 - Les incohérences devront être fournies au responsable de synchronisation pour qu'il puisse avoir un retour en temps réel et proposer un arbitrage si nécessaire.

Exemple d'application de la méthode 5 : Evaluation de l'adéquation du besoin avec l'approche

Les principes de synchronisation sont étudiés à la Table 12. Le PS1 sera prise en compte dans la méthodologie durant les étapes « Définitions des contextes d'ingénierie » et « Configuration de la synchronisation ». Le principe de synchronisation PS2 sera pris en compte à l'application de la synchronisation.

Table 12 Principes spécifiques définis par le responsable de synchronisation

Activités	**Principes d'interactions**	**Définitions des contextes d'ingénierie**	**Configurations de la synchronisation**	**Application de la synchronisation**	**Suivi, évolution des cohérences**
PS1		Durant la description des processus et des méthodes, identifier les jalons existants.	Définir les activités, les méthodes et les points de vue liés à des jalons dans les processus.		
PS2				A chaque itération, remonter l'information au responsable de synchronisation.	

3. ACTIVITE DE DEFINITION DES CONTEXTES DES DISCIPLINES D'INGENIERIE

3.1. OBJECTIF

L'activité « Définition des contextes d'ingénierie » de la méthodologie définit, de façon détaillée, les disciplines d'ingénierie et leurs contextes. Dans un premier temps, elle définit les processus et les activités, les méthodes et les points de vue associés aux disciplines d'ingénierie. Dans un second temps, elle identifie de possibles besoins de synchronisation.

Cette activité utilise les concepts et les modèles présentés à l'étape « définition des disciplines d'ingénierie » (cf. Chapitre II2).

3.2. METHODES

Les méthodes emploient les livrables d'entrée suivants :

- La liste des disciplines d'ingénierie et les motivations du sponsor ;
- La description des rôles et missions des disciplines ;
- La liste des principes généraux retenus ;
- La liste des principes spécifiques.

Deux méthodes sont proposées : « Définition des contextes d'ingénierie » et « Définition des besoins de synchronisation ».

Cette activité produit les livrables suivants :

- Pour chaque discipline, la définition du contexte d'ingénierie ;
- L'ensemble des vues (modèles) associées à la définition des contextes d'ingénierie ;
- Les ensembles des processus, des activités, des méthodes et des points de vue ;
- L'ensemble des besoins de synchronisation ;
- Les définitions des besoins de synchronisation ;
- L'ensemble de vues (modèles) associées à la contextualisation des besoins de synchronisation.

3.2.1. DEFINITION DES CONTEXTES D'INGENIERIE

La méthode suit une approche « top-down », i.e. elle s'intéresse aux concepts du plus au moins abstraits. Elle définit les contextes des disciplines d'ingénierie intervenant sur le système. Ces disciplines sont choisies par le sponsor et le responsable de synchronisation. Le cheminement est le suivant :

1. Recueillir le processus appliqué pour chaque discipline d'ingénierie ainsi que les référentiels auxquels elle est contrainte ;
2. Pour chaque processus, construire sa structure, ses activités, leurs objectifs et leurs préoccupations, en utilisant une représentation graphique adaptée.
3. Pour chaque activité, fournir l'ensemble des méthodes pouvant répondre aux objectifs d'étude. Les méthodes seront caractérisées par les données d'entrées, intermédiaires et finales (cf. définitions activité et méthode Chapitre II2.2).
4. Pour chaque méthode, recueillir les points de vue associés aux représentations du système étudié.

Cette méthode apporte une définition des contextes. En utilisant la représentation graphique définie au Chapitre II2.3.2, la visualisation des contextes permet de donner un aperçu des processus, des méthodes appliquées et des points de vue attendus.

3.2.2. DEFINITION DES BESOINS DE SYNCHRONISATION

Cette méthode a pour objectif d'identifier et de définir un ensemble de besoins de synchronisation candidats. Au travers de cette méthode, plusieurs questions se posent :

- **Quels sont les besoins** de l'expert vis-à-vis d'une autre expertise ?
- **Pourquoi** ont-ils besoin d'interagir ?
- **Quand** a-t-on besoin de se synchroniser ?
- **Que veut-on** échanger (quels objets, quelles propriétés) ?

La méthode suit les étapes suivantes :

1. L'identification des potentiels besoins de synchronisation ;
   - Identifier les points de vue pouvant avoir des éléments ou des concepts proches à des niveaux d'abstraction raisonnables ;
   - Recommandation :
     - *Il est conseillé de mener des séances de « brainstorming » et de travailler avec les ingénieurs en charge des études pour faire émerger un maximum de relations entre les points de vue.*
     - *En réutilisant le point de vue des contextes d'ingénierie construit dans la méthode précédente, l'identification des besoins de synchronisation peut être facilitée. Cela peut permettre de construire un tableau à deux dimensions considérant un certain nombre de disciplines d'ingénierie.*
     - *Certains points de vue très amont dans les processus ont probablement des dépendances avec de nombreux points de vue produits plus en aval dans les processus. Cependant, il faut s'interroger sur la pertinence de ces relations. Sont-elles pertinentes par rapport aux déroulés des processus, au niveau de raffinement et d'abstraction de ces points de vue ?*
2. La formalisation de chaque besoin de synchronisation ;
   - Fournir la motivation collective du besoin d'interaction, indépendamment de chaque discipline ;
   - Recueillir l'intérêt de ce besoin pour chaque discipline d'ingénierie ;
   - Recommandation :
     - *Cette étape peut être réalisée avec les praticiens. Le responsable de synchronisation doit étudier la pertinence d'une discipline, de mener cette interaction avec d'autres disciplines concernées ;*
   - Sélectionner les processus, les activités, les méthodes et les points de vue impliqués dans cette interaction.
   - Recommandation :
     - *Le besoin de synchronisation met en relation des points de vue issus de la « définition des contextes d'ingénierie ».*
   - Identifier, pour chaque point de vue, les éléments et les propriétés qui interviendront dans cette interaction.
3. La validation et l'élimination ;
   - Reprendre l'ensemble des besoins de synchronisation identifiés, vérifier et valider leur pertinence ;
   - Eliminer les besoins redondants. Si plusieurs besoins de synchronisation ont le même objectif à des niveaux d'abstraction identiques, mais avec des points de vue différents, il peut être pertinent de n'en considérer qu'un seul ;

- Recommandation :
  - *Note : Certaines questions peuvent être posées telles que « Ne pourrait-on pas mener cette interaction plus tôt dans les processus ? », « Est-ce que des besoins de synchronisation sont dépendants entre eux ? ».*
- Tracer les dépendances des besoins de synchronisation. Certains besoins peuvent avoir une dépendance ou être conditionnés par rapport à un autre besoin selon plusieurs critères comme :
  - la dépendance entre les vues ;
  - la logique organisationnelle des activités dans les processus.
- Recommandation :
  - *Une matrice de dépendance entre les besoins de synchronisation peut être effectuée.*

En utilisant la représentation graphique définie au Chapitre II2.3.3, la visualisation des contextes et des besoins de synchronisation offre un aperçu des dépendances entre les points de vue.

3.3. EXEMPLES D'APPLICATION

Le cas d'étude se veut générique et synthétique. Il contient des activités de très haut niveau. Ici, en Figure 61, les disciplines d'ingénierie sont l'architecture système à gauche et la sûreté de fonctionnement dans le secteur aéronautique (ARP 4761 [21]) à droite.

En utilisant la représentation graphique définie Chapitre II2.3.2, sont modélisés le processus, les méthodes et les points de vue d'une équipe d'architecture système d'une société fictive. De même, en sûreté de fonctionnement, le contexte a été reconstruit.

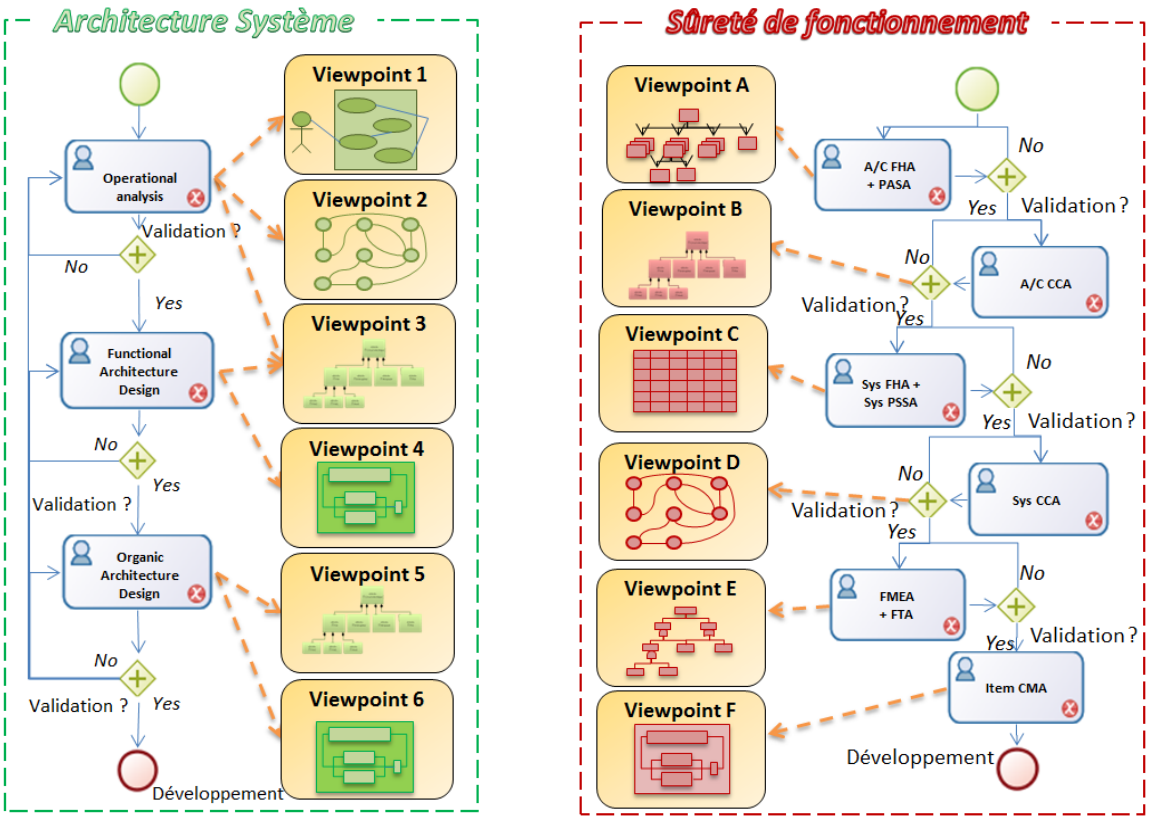


Figure 61 Exemples de vues des contextes de disciplines d'ingénierie

A partir de la figure précédente, des besoins de synchronisation sont identifiés. Dans cet exemple, trois besoins de synchronisation sont identifés :

- Besoin de synchronisation 1 : Mise en cohérence des états opérationnels
- Besoin de synchronisation 2 : Mise en cohérence des fonctions
- Besoin de synchronisation 3 : Mise en cohérence des composants du système

Ils sont illustrés dans la Figure 62 et détaillés dans la Table 13.

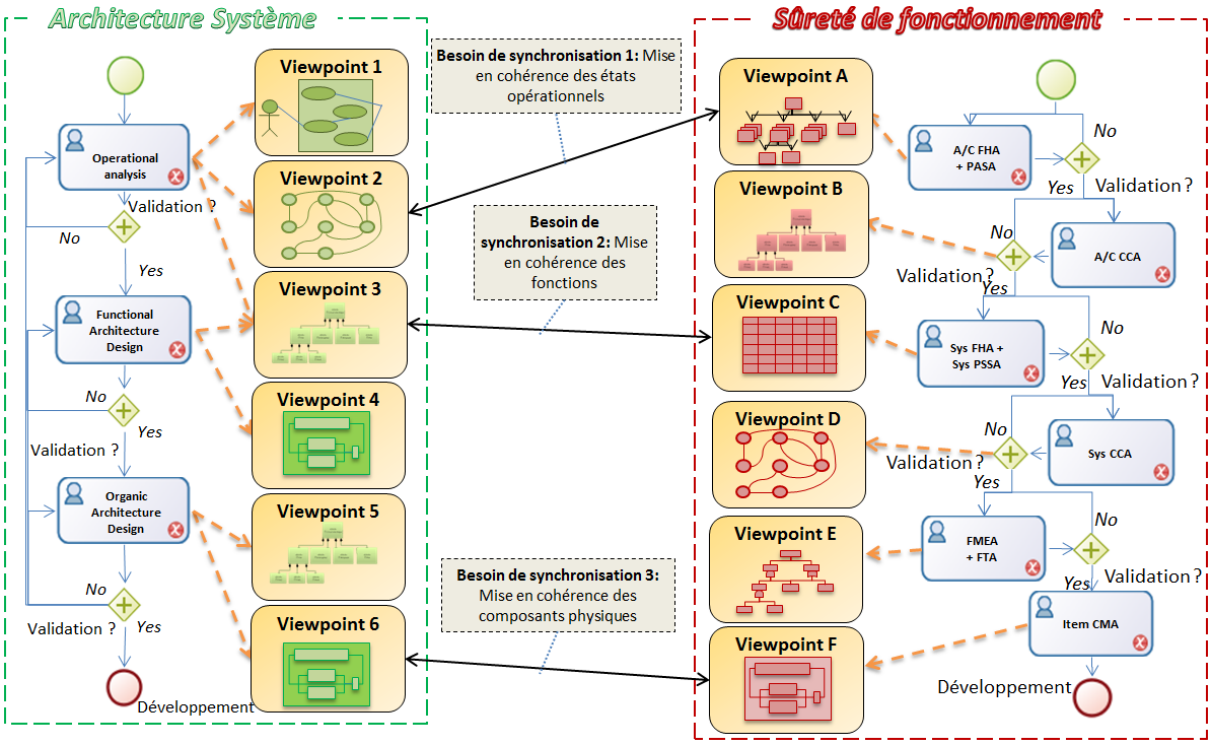


Figure 62 Exemple de vues contextualisées des besoins de synchronisation

Table 13 Exemple de description des besoins de synchronisation

Nom	Besoin de l'architecture système vis-à-vis de la sûreté de fonctionnement	Besoin de la sûreté de fonctionnement vis-à-vis de l'architecture système	Raison de ce besoin	Point de vue du 1er contexte / Point de vue du 2nd contexte	Les éléments, propriétés et/ou relations mise en jeu (Architecture puis sûreté de fonctionnement)
Besoin de synchronisation 1	Vérifier que des états n'ont pas été oubliés.	Tenir compte du découpage des états dans les études de sûreté.	Mise en cohérence des états opérationnels	Machine à états	Machines à états, états, transition entre états
				Décomposition hiérarchique des états	Etat et relation de composition
Besoin de synchronisation 2	Communiquer les fonctions et faire valider leur architecture.	Identifier les évènements redoutés liés à la perte de fonction.	Mise en cohérence des fonctions	Diagramme de Définition de block - SysML	Block, Relation de composition,…
				Tableau FHA	Fonction, relation de composition, Conditions de panne, …
Besoin de synchronisation 3	Communiquer les composants et faire valider leur architecture.	Evaluer l'architecture selon les évènements redoutés identifiés.	Mise en cohérence des composants du système	Internal Block Diagram - SysML	Block, Part, Connector, Port, …
				Modèle AltaRica 3.0	Block, Class, Assertion,…

La Table 13 montre d'ores et déjà des dépendances entre les besoins de synchronisation. En, effet les éléments concernés par les besoins de synchronisation peuvent avoir des dépendances entre eux lors du déroulement du processus de synchronisation, notamment les états opérationnels avec les fonctions, les fonctions avec les composants du système, etc.

4. ACTIVITE DE CONFIGURATION DE SYNCHRONISATION

4.1. OBJECTIF

L'activité « Configuration de synchronisation » de la méthodologie permet de définir et de configurer des « points de synchronisation » à partir des besoins de synchronisation. Elle permet de générer l'application de la synchronisation, c'est-à-dire la réalisation des synchronisations entre les vues des ingénieurs concernés. Elle identifie, formalise et organise les interactions à mener.

Cette activité utilise les concepts et les modèles présentés à l'étape « Définition des concepts d'architecture » (cf. Chapitre II3) et « Configuration des interactions multidisciplinaires » (cf. Chapitre II4).

4.2. METHODE DE FORMALISATION DES POINTS DE SYNCHRONISATION

La méthode proposée ci-dessous permet de définir l'ensemble des points de synchronisation, de les caractériser, de les ordonnancer et de définir les bibliothèques nécessaires.

Elle emploie les livrables d'entrée suivants :

- La définition des contextes d'ingénierie ;
- L'ensemble des vues (modèles) associées à la définition des contextes d'ingénierie ;
- Les ensembles des processus, des activités, des méthodes et des points de vue ;
- L'ensemble des besoins de synchronisation ;
- La définition des besoins de synchronisation ;
- L'ensemble des vues (modèles) associées à la contextualisation de besoin de synchronisation.

La **formalisation des points de synchronisation** se déroule en quatre étapes successives :

1. L'identification des points de synchronisation :
   - Pour chaque besoin de synchronisation, identifier un ou plusieurs points de synchronisation nécessaire.
   - Remarque :
     - *Il peut y avoir plusieurs points de synchronisation par besoin de synchronisation si l'on veut séparer les préoccupations de l'architecture du système en plusieurs étapes.*
2. La construction des bibliothèques :
   - Construire une bibliothèque des concepts pivots d'architecture. Une bibliothèque par défaut est fournie ;
   - Construire une bibliothèque contenant des listes de compromis clustérisés par contexte. Une bibliothèque par défaut est fournie.
3. La définition des points de synchronisation :
   - Sélectionner le ou les concepts pivots d'architecture à considérer par le point de synchronisation dans la bibliothèque correspondante ;

- Sélectionner une ou plusieurs listes de compromis dans la bibliothèque correspondante pour chaque discipline d'ingénierie concernée par le point de synchronisation ;
- Construire les mappings (ou réutiliser/adapter des mappings existants) pour spécifier le passage de chaque point de vue vers le ou les concept(s) pivot(s) d'architecture.

4. L'ordonnancement des points de synchronisation :
   - En reprenant les points de synchronisation issus d'un même besoin de synchronisation, définir si nécessaire, l'ordre dans lequel ils doivent être appliqués ;
   - En reprenant l'ensemble des points de synchronisation, spécifier les contraintes d'ordonnancement au regard des processus et de l'ordonnancement logique des activités ;
   - Allouer à chaque point de synchronisation, les contraintes pré-requises pour son application.

Cette activité produit les livrables suivants :

- Pour chaque besoin de synchronisation, l'ensemble des points de synchronisation associés ;
- Les bibliothèques de concepts pivots d'architecture et des listes de compromis ;
- Les définitions des points de synchronisation ;
- L'ordonnancement des points de synchronisation avec leurs préconditions.

4.3. EXEMPLES D'APPLICATION

L'exemple est repris de la section précédente (cf. 3.3). A partir du besoin de synchronisation 1 « Mise en cohérence des états opérationnels du système », deux points de synchronisation ont été définis. Le premier est dédié à la sémantique des éléments et le second à la composition / hiérarchie des états.

Une bibliothèque des concepts pivots d'architecture est construite. Elle contient deux concepts de structuration (cf. Figure 63) :

- Le concept d'élément conteneur de type « état » ;
- Le concept de composition.

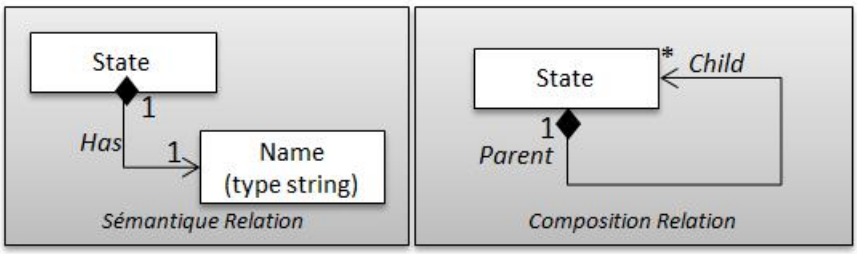


Figure 63 Concept d'objet sémantique et relation de composition

Une bibliothèque des compromis a également été construite. Ici, les compromis sont les mêmes pour les deux disciplines d'ingénierie :

- Pour le point de synchronisation 1, les actions « Renommer », « Ajouter » et « Supprimer » un état ;
- Pour le point de synchronisation 2, les actions « Ajouter », « Supprimer » une relation de composition et « Déplacer » un état.

Les Figure 64 et Figure 66 décrivent les deux points de synchronisation. Les mappings sont explicites dans les Figure 65 et Figure 67.

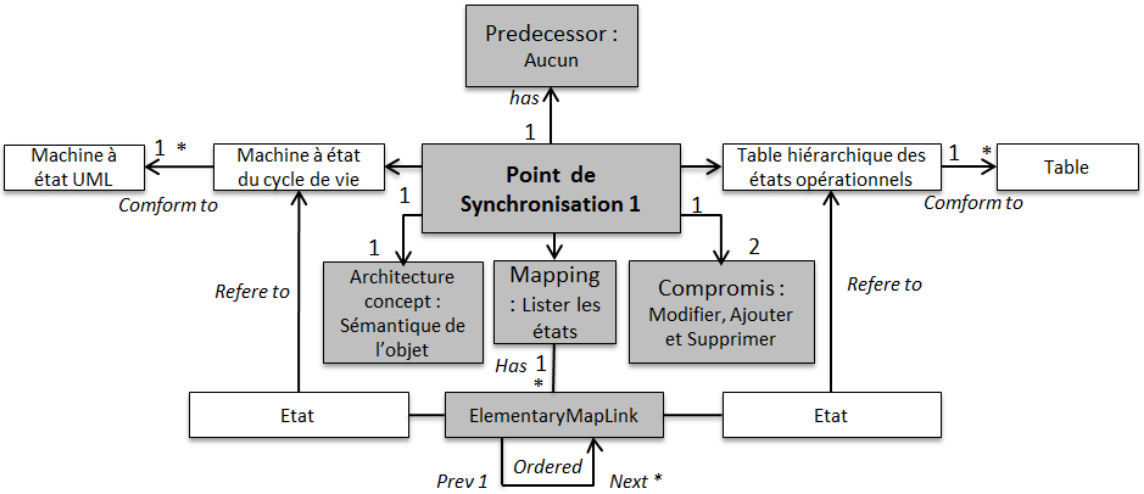


Figure 64 Définition du point de synchronisation 1

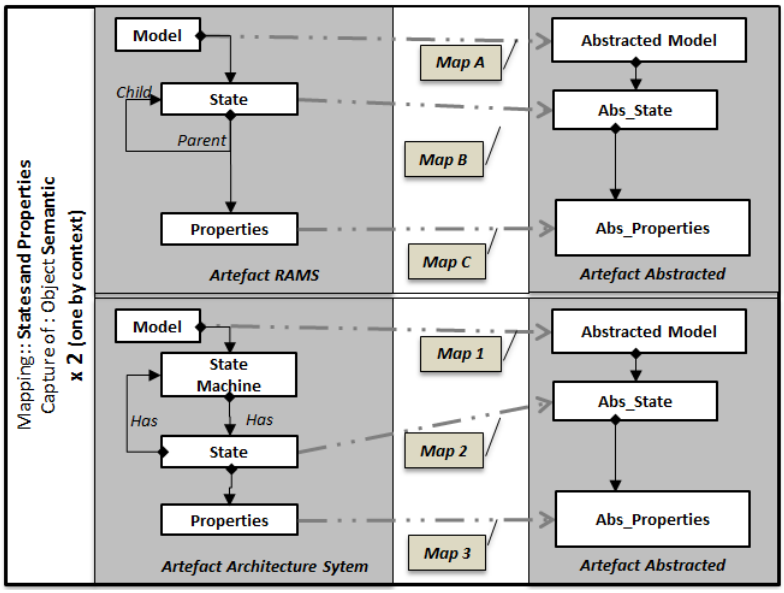


Figure 65 Mapping du point de synchronisation 1

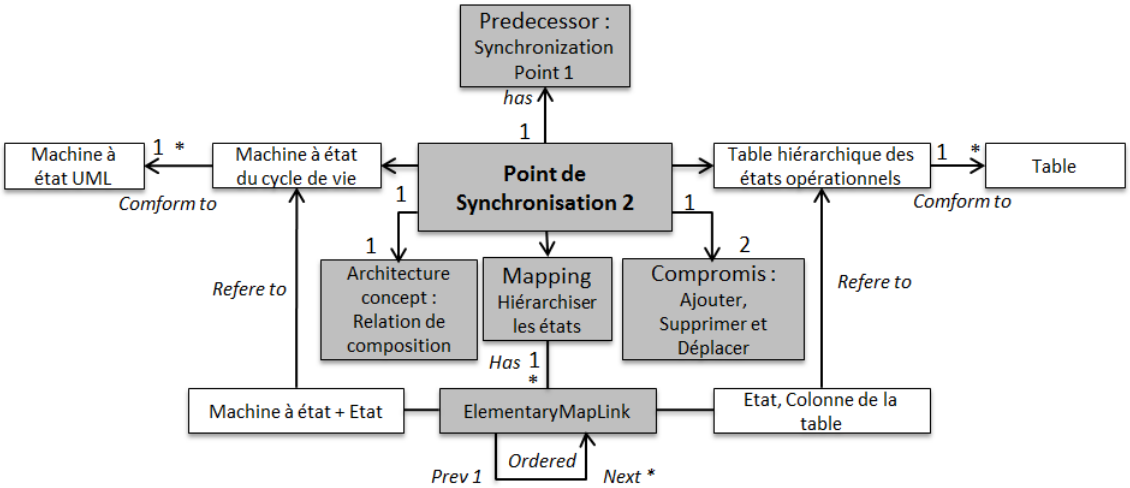


Figure 66 Définition du point de synchronisation 2

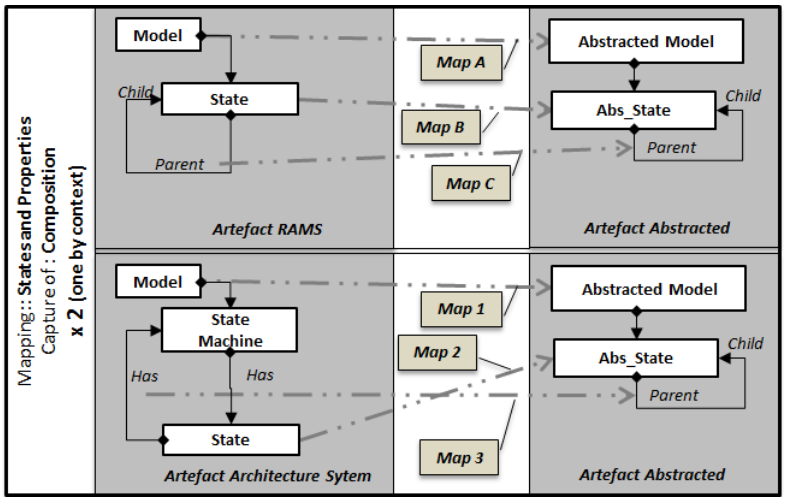


Figure 67 Mappings du point de synchronisation 2

L'ordonnancement des points de synchronisation est pris en charge par la propriété « Predecessor ». Ici Figure 68, le point de synchronisation 2 ne peut être effectué avant la finalisation du point de synchronisation 1.

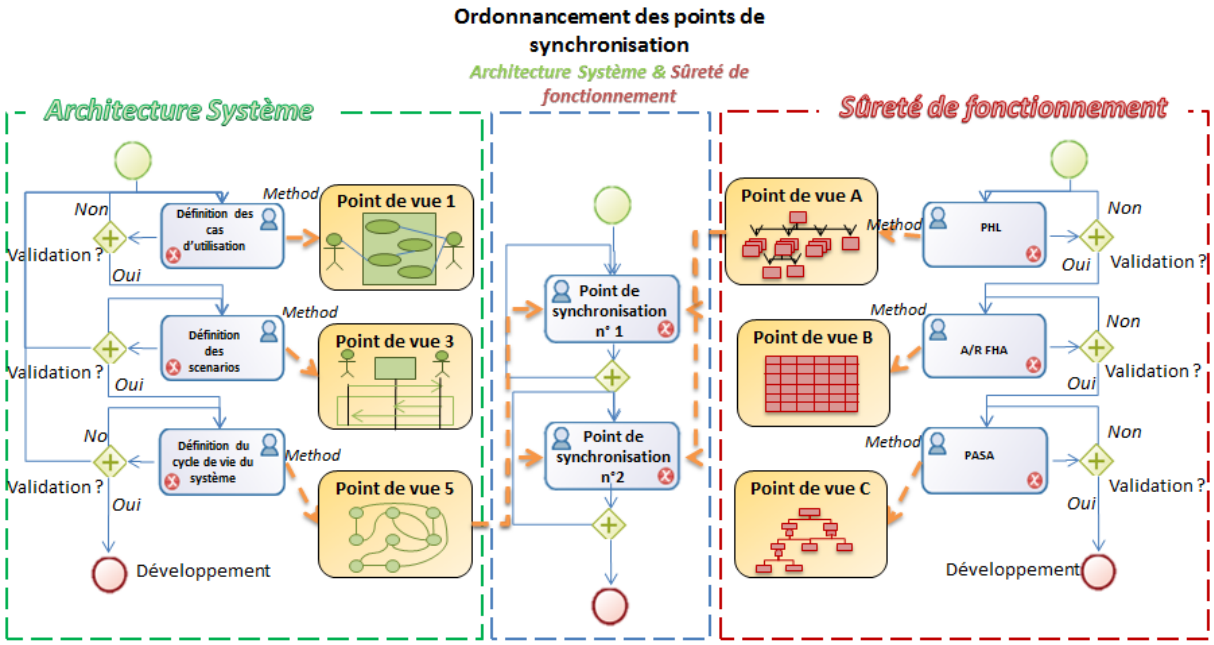


Figure 68 Ordonnancement des points de synchronisation

5. ACTIVITE D'APPLICATION DE SYNCHRONISATION

5.1. OBJECTIF

L'activité « Application de la synchronisation » de la méthodologie explique comment l'exécuter et satisfaire les paramètres définis par la configuration de synchronisation.

L'activité réutilise les concepts et les modèles présentés à l'étape « Mise en cohérence » (cf. Chapitre II5) et « Application de la synchronisation de modèles » (cf. Chapitre II6).

L'application doit vérifier et maintenir une séparation des préoccupations entre la conception d'architecture système et les analyses de sûreté de fonctionnement. Les techniques de synchronisation de modèles sont rappelées ci-dessous :

Synchronisation = Abstraction + Comparaison + Concrétisation

5.2. METHODE APPLICATIVE DE SYNCHRONISATION DE MODELES

Elle emploie les livrables d'entrée suivants :

- La définition du contexte d'ingénierie ;
- L'ensemble des vues (modèles) associées à la définition des contextes d'ingénierie ;
- Les ensembles des processus, des activités, des méthodes et des points de vue ;
- L'ensemble des points de synchronisation associés ;
- Les bibliothèques de concepts pivots d'architecture ;
- Les bibliothèques des compromis ;
- Les définitions des points de synchronisation ;
- L'ordonnancement des points de synchronisation avec leurs préconditions.

La méthode proposée est décomposée en 7 étapes :

1. La validation des relations de cohérence sur les vues concernées : Elle consiste à vérifier la validité des relations de cohérence issues d'itérations ou de points de synchronisation antérieures. (Action automatique)
2. L'abstraction (Action automatique)
3. La comparaison (étapes itératives, tant que tous les éléments n'ont pas été traités)
    1. Comparaison des éléments des modèles (Action automatique)
    2. Formulation des questions aux ingénieurs (Action automatique)
    3. Interprétation des réponses (Action automatique)
    4. Synthèse des relations de cohérence et des incohérences (Action automatique).
4. La proposition de compromis (Action semi-automatique)
5. La concrétisation : Annotation des vues des contextes par les compromis choisis (manuelle).
6. L'évolution des vues dans chaque contexte. (manuelle), puis lancement d'une nouvelle itération (retour à l'étape 1 du dessus).

Exception : Si à l'étape 3.4, aucune incohérence n'a été détectée, alors tous les éléments retrouvent leur équivalence. Aucun compromis n'est demandé (ne plus réaliser les étapes 4, 5 et 6). On passe directement à l'étape 7.

7. L'enregistrement des relations de cohérence dans un fichier de traces dédié au point de synchronisation. Ce fichier cartographie les relations de cohérence des éléments des vues de chaque contexte. Si des incohérences ont étés constatées dans des itérations antérieures, alors celles-ci seront rattachées à la relation de cohérence concernée. L'historique des évolutions et des discussions entre des praticiens est ainsi capitalisé.

Voici en Figure 69, un diagramme BPMN qui résume les étapes de la méthode :

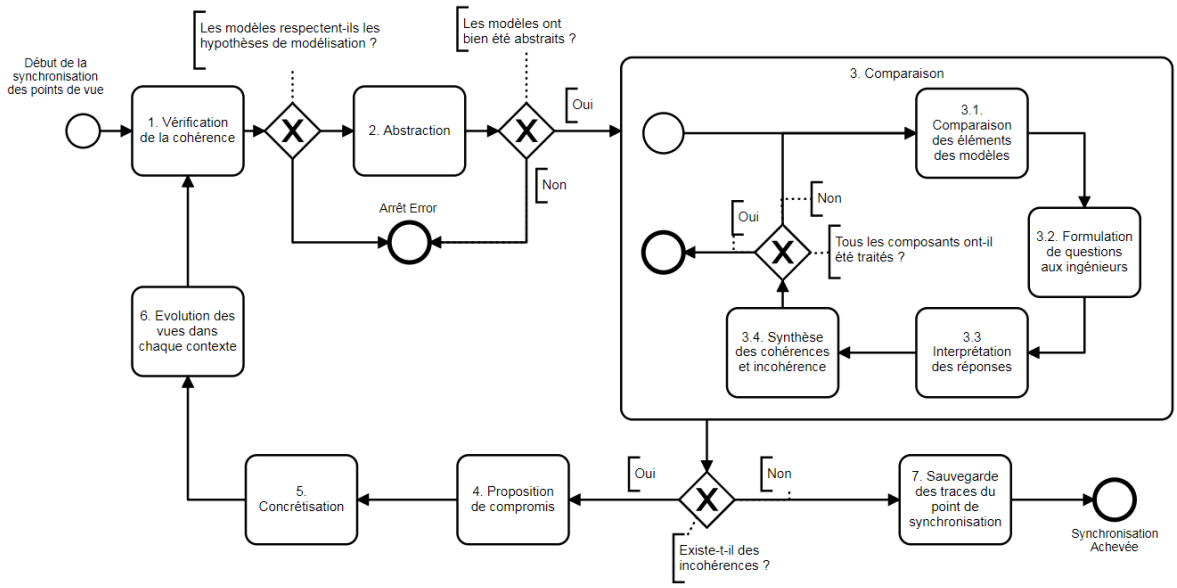


Figure 69 Méthode de synchronisation de modèles – BPMN

L'activité produit les livrables suivants :

- Ensemble de modèles de cohérence ;
- Liste de toutes les incohérences ;
- Liste de toutes les relations de cohérence ;
- Liste des compromis appliqués ;
- Nombre d'itérations effectuées.

5.3. EXEMPLES D'APPLICATION

L'exemple présenté est un cas simple de synchronisation durant les étapes très amont du cycle de développement du système. Il illustre des vues et les résultats des différentes étapes de la méthode de synchronisation. Le point de synchronisation concerne, ici, les états du cycle de vie d'un système hélicoptère. Il s'intéresse à la composition des états.

Trois vues sont issues de l'architecte système, elles sont représentées en Figure 70 et Figure 71

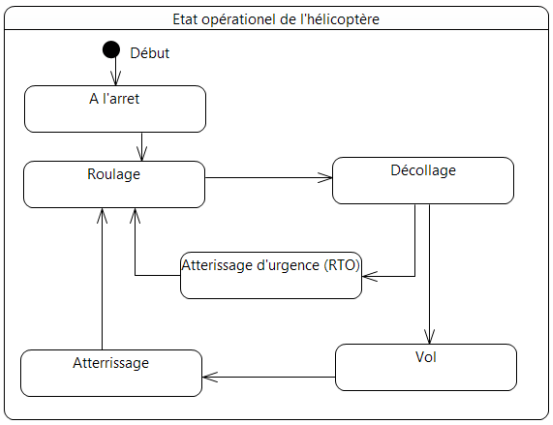


Figure 70 Machine à états UML de l'état opérationnel de l'hélicoptère – Vue de l'architecture système

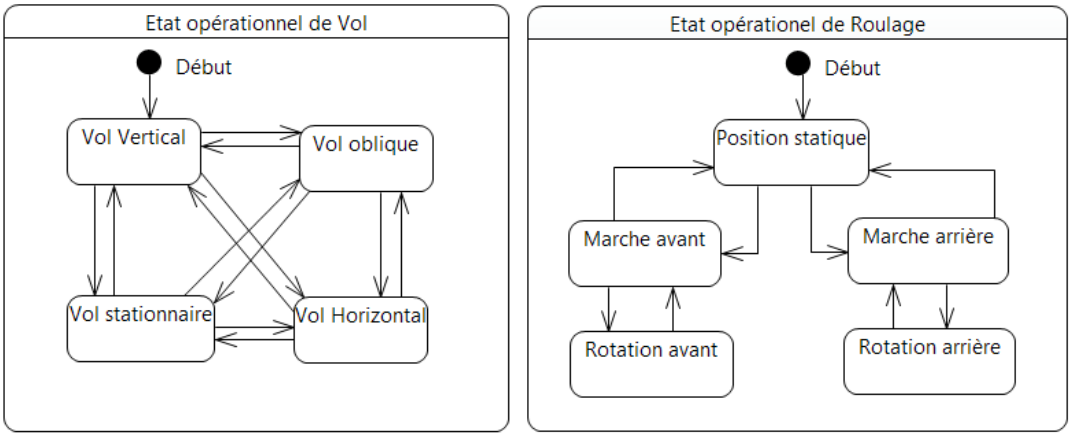


Figure 71 Machines à états du Vol et du Roulage – Vues de l'architecture système

Une vue est issue de la sûreté de fonctionnement à la Figure 72. Elle hiérarchise les états du cycle de vie du système. Il s'agit de données d'entrée de la méthode Aircraft FHA dans l'ARP 4754 [131].

Etats (lvl1)	**Etats (lvl2)**
Arrêt	
Roulage	
Décollage	Décollage optimal, Décollage avec un moteur défaillant (OEI)
Atterrissage d'urgence (RTO)	
Vol	Vol vertical, Vol oblique, Vol stationnaire, Vol horizontal
Atterrissage	Atterrissage optimal, Atterrissage avec un moteur défaillant (OEI)

Figure 72 Liste hiérarchisée des états opérationnels du système – Vue de la sûreté de fonctionnement

Les résultats de l'abstraction sont les vues représentées Figure 73 et Figure 74.

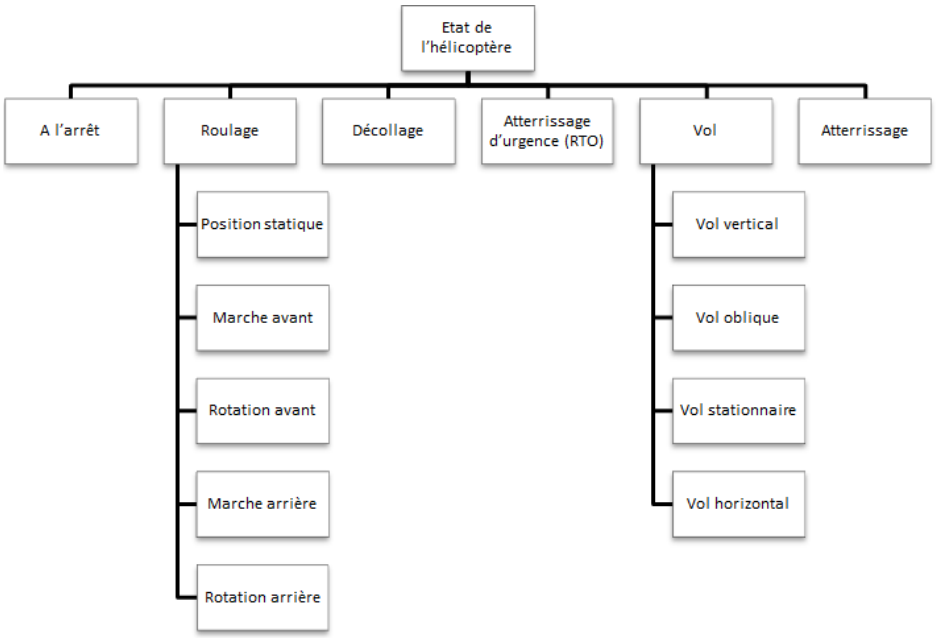


Figure 73 Abstraction des états – Vue de l'architecture système

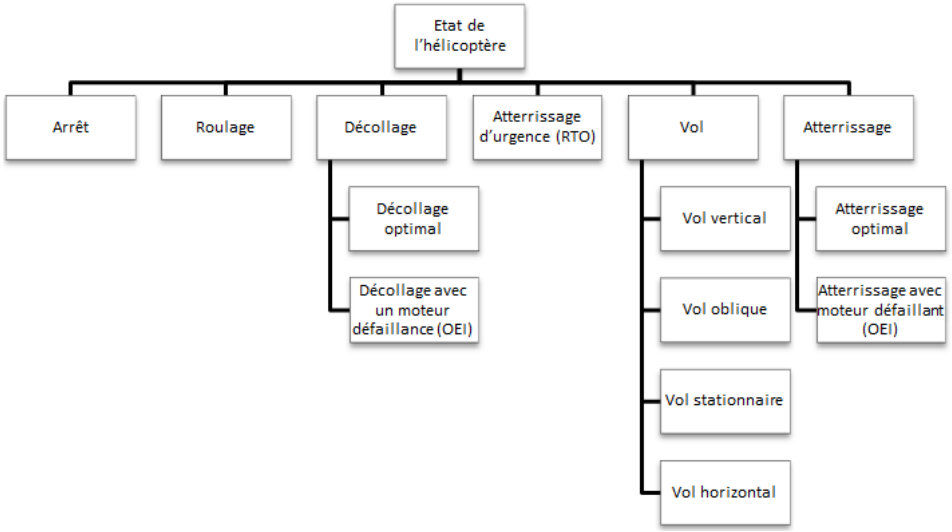


Figure 74 Abstraction des états – Vue de la sûreté de fonctionnement

Résultat de comparaison des abstractions. A partir de la saisie par des ingénieurs, trois modèles de cohérence sont produits en fonction du niveau hiérarchique des états. Les modèles de cohérence tracent les relations de cohérence et déduisent les incohérences liées à la composition des états du système. Au total, 11 relations de cohérence et 9 incohérences seront construites.

La Figure 75 restitue le résultat de la comparaison entre les éléments du premier niveau hiérarchique du contexte ingénierie système et contexte sûreté de fonctionnement.

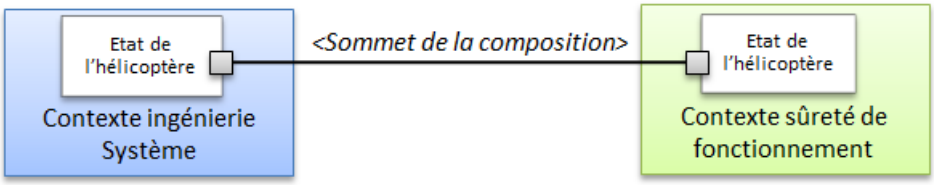


Figure 75 Vue des relations de cohérence du premier niveau hiérarchique des abstractions

La Figure 76 restitue le résultat de la comparaison entre les éléments du second niveau hiérarchique du contexte ingénierie système et contexte sûreté de fonctionnement.

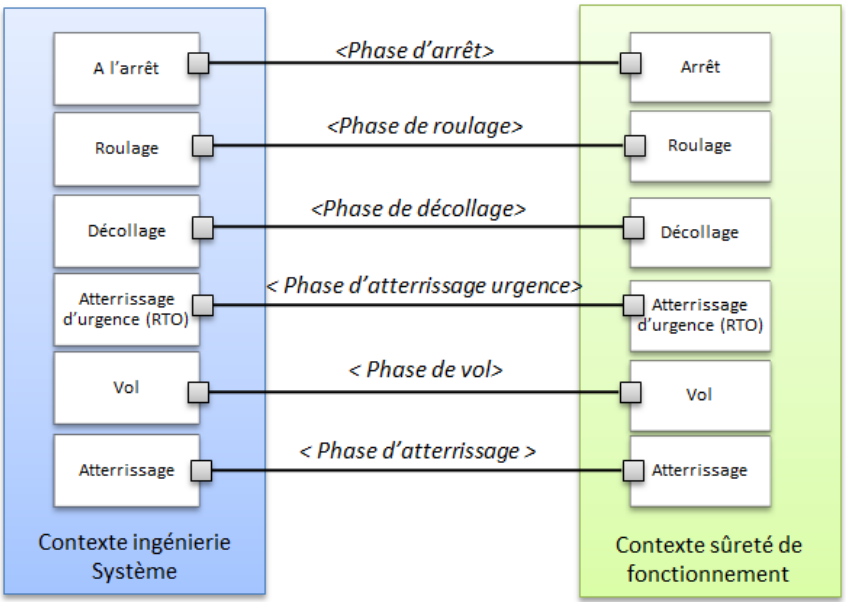


Figure 76 Vue des relations de cohérence du second niveau hiérarchique des abstractions

La Figure 77 restitue le résultat de la comparaison entre les éléments du troisième niveau hiérarchique du contexte ingénierie système et contexte sûreté de fonctionnement.

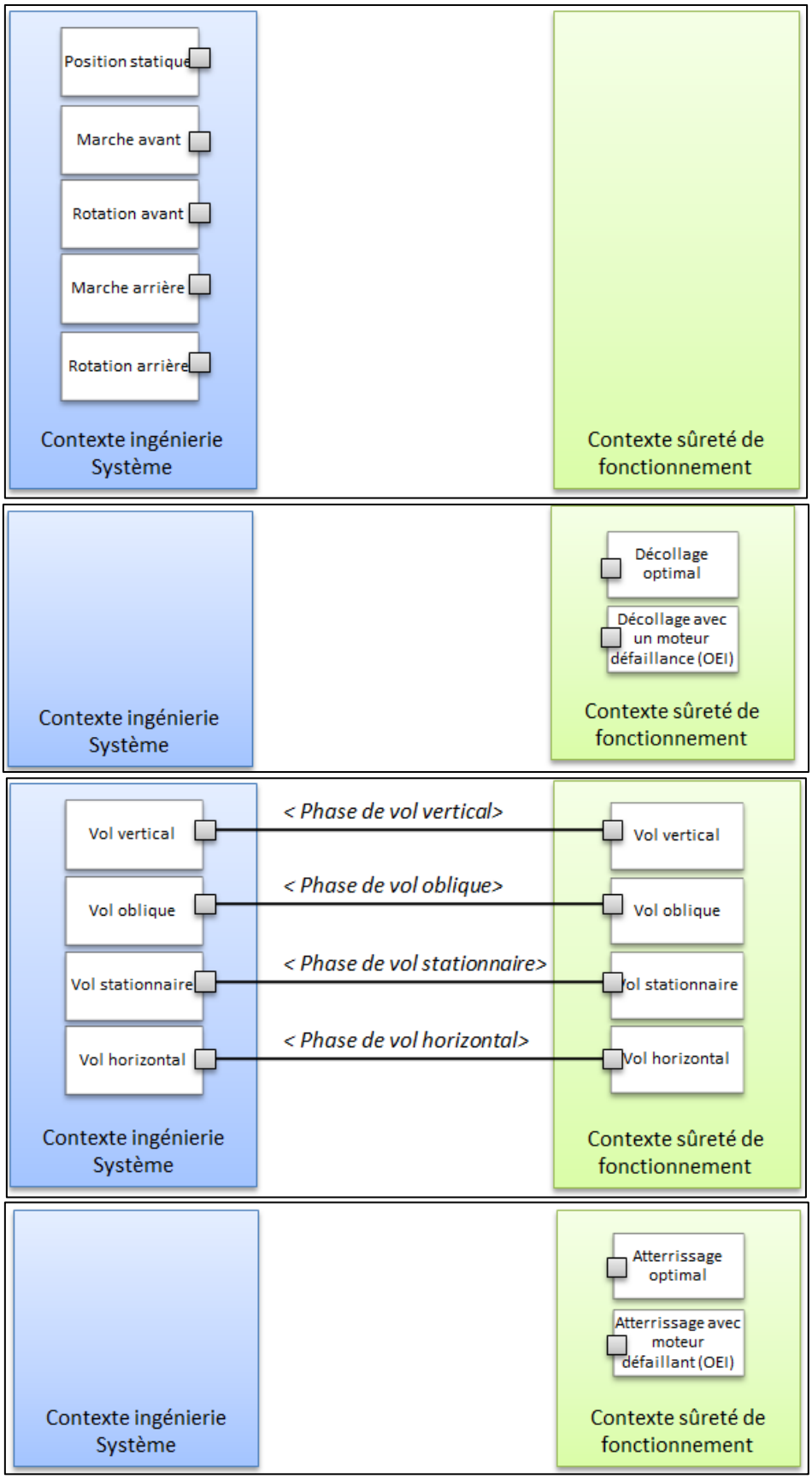


Figure 77 Vues des relations de cohérence du troisième niveau hiérarchique des abstractions

Résultat de concrétisation :
Pour chaque incohérence, un ensemble de compromis est proposé : Ajouter un élément, Supprimer un élément, Déplacer un élément, Renommer un élément, Exclure l'élément de l'interaction (i.e. ne pas tenir compte de cet élément). Les ingénieurs sélectionnent le ou les compromis qu'il semble judicieux d'appliquer.

Liste des incohérences et compromis choisis :

- « Position statique » n'a pas d'équivalence dans le contexte sûreté de fonctionnement. Les ingénieurs proposent de ne pas tenir compte de cet état ;
- « Marche avant » n'a pas d'équivalence dans le contexte sûreté de fonctionnement. Les ingénieurs proposent de ne pas tenir compte de cet état ;
- « Rotation avant » n'a pas d'équivalence dans le contexte sûreté de fonctionnement. Les ingénieurs proposent de ne pas tenir compte de cet état ;
- « Marche arrière » n'a pas d'équivalence dans le contexte sûreté de fonctionnement. Les ingénieurs proposent de ne pas tenir compte de cet état ;
- « Rotation arrière » n'a pas d'équivalence dans le contexte sûreté de fonctionnement. Les ingénieurs proposent de ne pas tenir compte de cet état ;
- « Atterrissage optimal » n'a pas d'équivalence dans le contexte ingénierie système. Les ingénieurs proposent d'ajouter cet état à la vue de l'architecture système ;
- « Atterrissage avec moteur défaillant (OEI) » n'a pas d'équivalence dans le contexte ingénierie système. Les ingénieurs proposent d'ajouter cet état à la vue de l'architecture système ;
- « Décollage optimal » n'a pas d'équivalence dans le contexte ingénierie système. Les ingénieurs proposent d'ajouter cet état à la vue de l'architecture système ;
- « Décollage avec un moteur défaillant (OEI) » n'a pas d'équivalence dans le contexte ingénierie système. Les ingénieurs proposent d'ajouter cet état à la vue de l'architecture système.

Ainsi, les nouvelles vues des deux disciplines sont celles représentées Figure 78 et Figure 79.

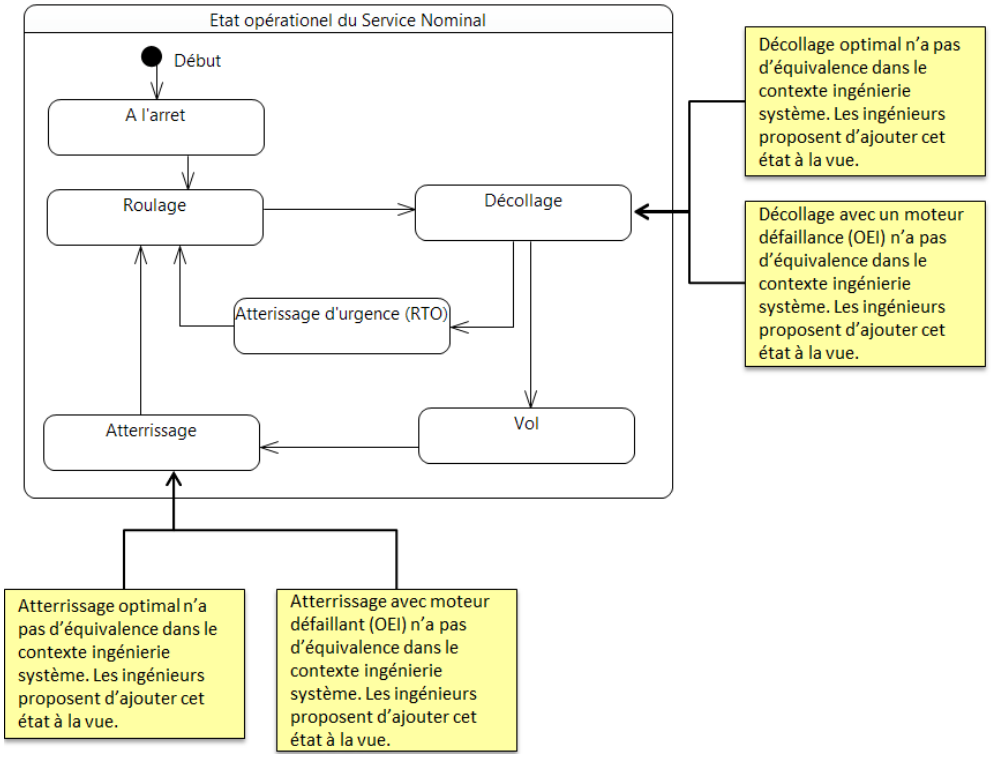


Figure 78 Vue concrétisée après itération d'une synchronisation sur la machine à état de l'hélicoptère

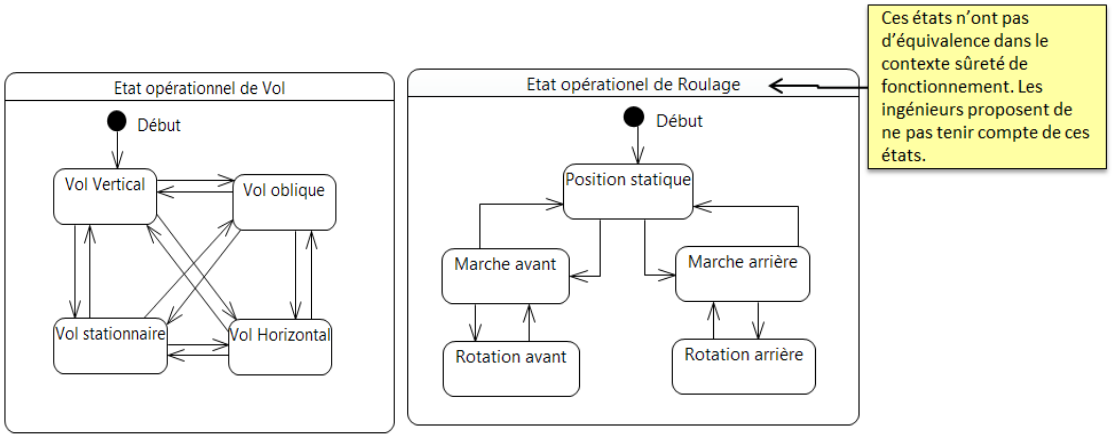


Figure 79 Vues concrétisées après itération d'une synchronisation sur les machines à états des états roulage et Vol

Dans cet exemple, les compromis se concrétisent tous sur la vue de l'architecte système. Mais, ceci n'est pas généralisable.

Mise à jour des vues :
L'architecte système a alors ajouté deux machines à états, pour décliner les états « Décollage » et « Atterrissage ». Elles sont présentées à la Figure 80.

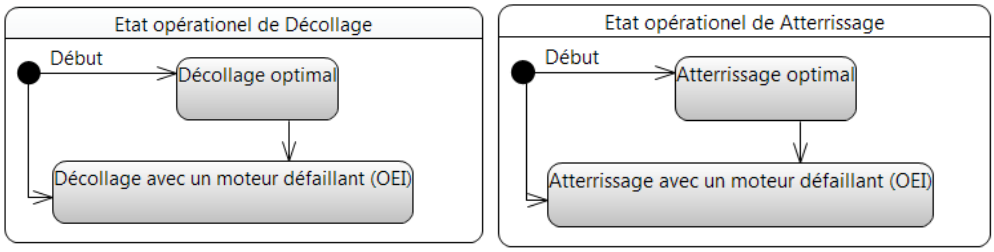


Figure 80 Vues construites en complément par l'architecte système après la première itération

Les ingénieurs peuvent réitérer la synchronisation pour espérer se mettre d'accord après la mise à jour des vues.

6. ACTIVITE DE SUIVI DE LA COHERENCE ET EVOLUTION DE LA SYNCHRONISATION

6.1. OBJECTIF

L'activité « Application de la synchronisation » de la méthodologie permet, dans un premier temps, de capitaliser les liens de traçabilité entre les éléments manipulés. Dans un second temps, elle fait ressortir des informations sur l'exécution de l'activité « Application de la synchronisation ».

Les traces construites permettent de déduire et d'alimenter le retour d'expérience. Ce dernier est utilisé pour évaluer la solution implémentée et/ou formuler une demande d'évolution auprès du sponsor et du responsable de la synchronisation.

L'activité réutilise les concepts et les modèles présentés aux étapes « Mise en cohérence » (cf. Chapitre II5), « Application de la synchronisation de modèles » (cf. Chapitre II6) et « Traçabilité et histoire des modèles ».

6.2. METHODES ENVISAGEES

Les méthodes emploient les livrables d'entré suivants :

- Ensemble de modèles de cohérence ;
- Liste de toutes les incohérences ;
- Liste de toutes les relations de cohérence ;
- Liste des compromis appliqués ;
- Nombre d'itérations effectuées.
- Définitions des contextes d'ingénierie ;
- Ensembles des processus, des activités, des méthodes et des points de vue.

Deux méthodes sont proposées : la « Capitalisation des traces » et l'« Interprétation des traces ».

L'activité produit les livrables suivants :

- Les éléments de traçabilité :
  - de synchronisation (Globale) ;
  - d'un point de synchronisation ;
  - d'une itération ;
  - d'abstraction ;
  - de comparaison ;
  - de concrétisation.
- Des informations et des indicateurs sur l'histoire des synchronisations ;
- Des demandes d'évolution.

6.2.1. CONSTRUCTION DES TRACES

La méthode « construction des traces » construit l'ensemble des éléments de traçabilité. Elle intervient à plusieurs moments selon leur nature.

Les premiers éléments de traçabilité sont construits durant l'application de la synchronisation. Ils sont générés après l'abstraction, la comparaison et la concrétisation.

- La traçabilité d'abstraction est construite après chaque abstraction de modèles. Il y a autant d'éléments de traçabilité que de contextes concernés par l'interaction.

- La traçabilité de comparaison est construite après chaque comparaison. Il existe un seul élément de traçabilité par comparaison.

- La traçabilité de concrétisation est construite s'il existe au moins une incohérence et que les ingénieurs et architectes ont proposé un compromis à appliquer sur l'une des vues.

D'autres éléments de traçabilité sont construits à partir de ces premiers éléments, comme :

- La traçabilité d'une itération, qui est en quelques sortes un « package », i.e. un conteneur d'éléments qui capitalise les éléments suivants :

  - Les traçabilités d'abstraction (autant que de contextes d'ingénierie) ;
  - La traçabilité de la comparaison (une seule par itération) ;
  - La ou les traçabilité(s) de concrétisation (de 0 au nombre de contextes d'ingénierie).

- La traçabilité d'un point de synchronisation est aussi un « package » qui stocke toutes les traçabilités des itérations. Elle est construite lorsqu'un point de synchronisation est complètement traité, i.e. que la relation de cohérence traitée entre les vues est entièrement satisfaite.

- Enfin la traçabilité de synchronisation (Globale), qui est également un « package », qui capitalise toutes les abstractions, les comparaisons et les concrétisations effectuées dans toutes les itérations de tous les points de synchronisation. Cet élément de traçabilité suffit à lui seul pour résumer toutes les tâches conduites automatiquement ou non, durant la synchronisation de modèles. Il peut fournir l'ensemble des relations de cohérence et des incohérences étudiées.

6.2.2. INTERPRETATIONS DES TRACES

A partir de la traçabilité de synchronisation, il est possible de déduire des indicateurs et d'alimenter le retour d'expérience. Celui-ci est profitable à plusieurs parties prenantes :

- les ingénieurs et les architectes pour retrouver les justifications et enrichir la documentation ;
- le responsable de synchronisation pour évaluer l'efficacité les points de synchronisation traités ;
- le sponsor pour obtenir des indicateurs globaux.

La traçabilité permet de déduire des points durs pour faire évoluer la démarche. Toute l'histoire déroulée peut être reconstruite. On peut notamment faire ressortir les résultats suivants :

- Les concepts pivots d'architecture les plus utilisés ;
- Les incohérences les plus identifiées et traitées ;
- Les incohérences qui n'ont jamais pu être traitées ;
- Les relations de cohérence intuitives qui pourraient être éventuellement automatisées ;
- Une remise en question de la définition, de la configuration, de l'application de la synchronisation ;
- Une remise en question de la méthodologie de synchronisation.

Elle permet également de présenter des indicateurs au sponsor comme :

- Le gain et l'évolution sur le cycle de développement des projets ;
- Le nombre de points de synchronisation traités ;
- Le nombre d'itérations par points de synchronisation.

A partir des informations récupérées, le responsable de synchronisation avec les disciplines d'ingénierie peuvent formuler de nouvelles demandes d'évolution concernant :

- L'une des étapes du processus utilisées ;
- La méthodologie d'interactions multidisciplinaires ;
- La définition des contextes d'ingénierie ;
- La définition des points de synchronisation ;
- Les concepts pivots d'architectures ;
- Les techniques d'abstraction, de comparaison et de concrétisation.

6.3. EXEMPLES D'APPLICATION

Les exemples, présentés ci-dessous, réutilisent le cas d'application de la synchronisation présenté à la section 5.3.

La Figure 81 illustre les traces générées par l'abstraction des vues « Etat opérationnel de l'hélicoptère », « Etat opérationnel de vol » et « Etat opérationnel de Roulage ».

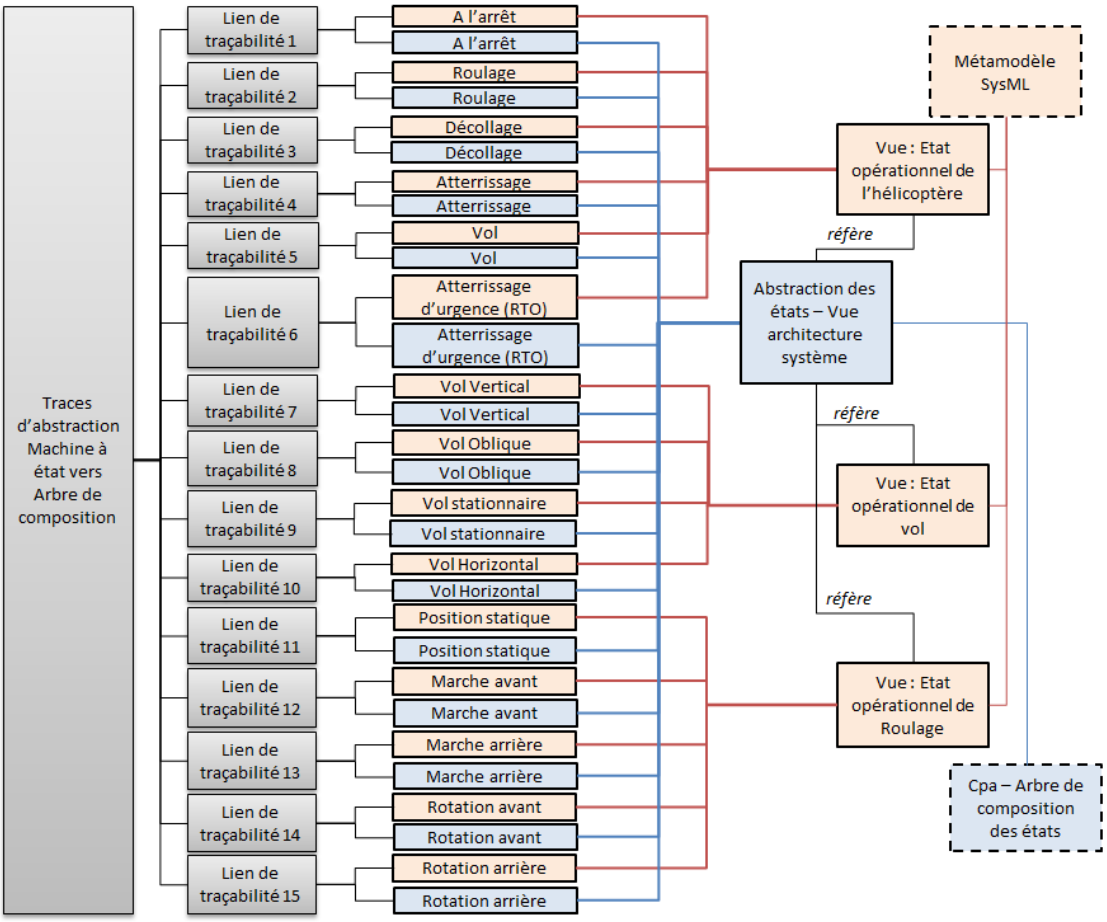


Figure 81 Exemple de la traçabilité d'une abstraction

La Figure 82 illustre les traces générées par la comparaison des abstractions des états. Elle indique clairement les liens de traçabilité liés à une relation de cohérence et les liens de traçabilité liés à une incohérence.

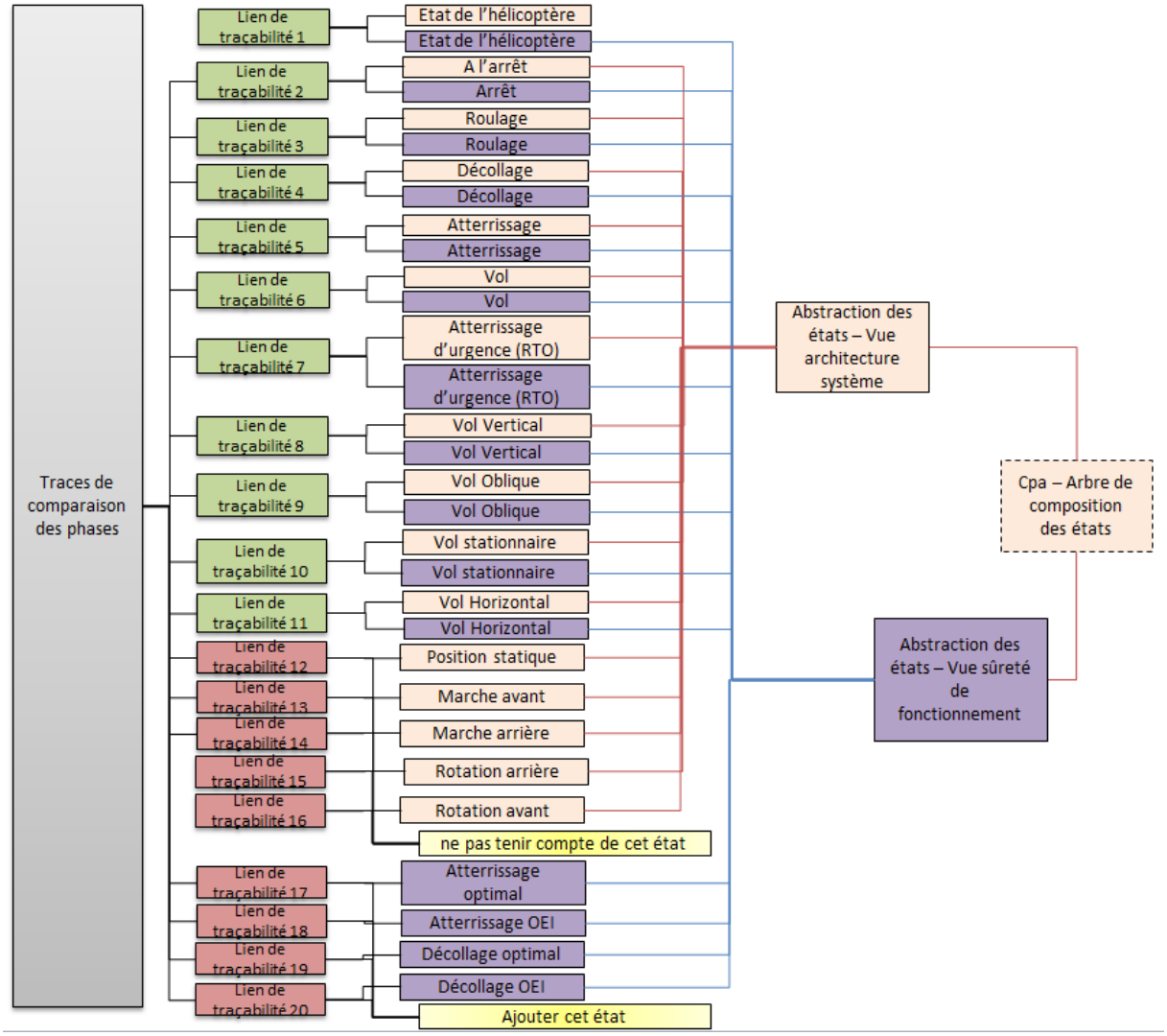


Figure 82 Exemple de la traçabilité d'une comparaison

La Figure 83 illustre les traces générées par la concrétisation des compromis sur les vues « Etat opérationnel de l'hélicoptère », « Etat opérationnel de vol » et « Etat opérationnel de Roulage ».

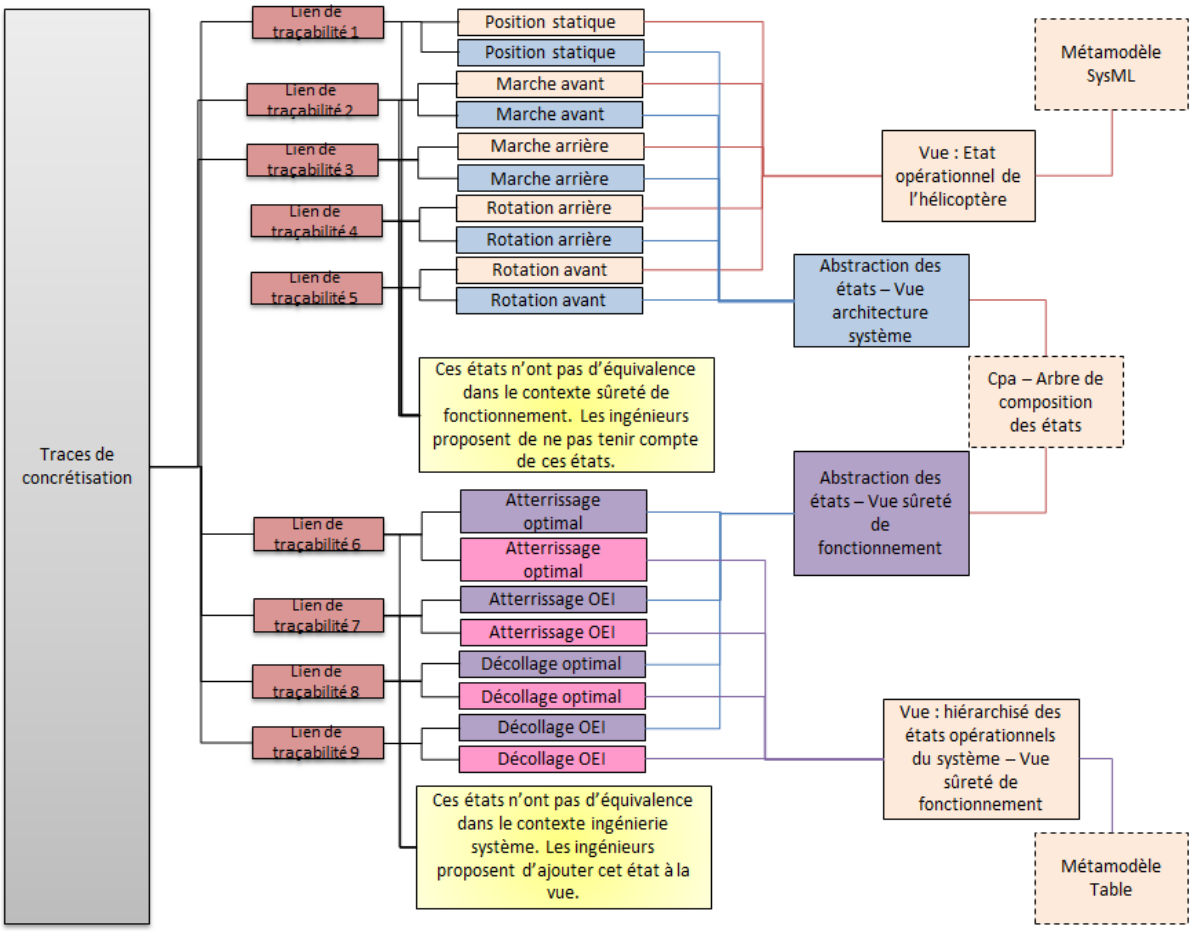


Figure 83 Exemple de la traçabilité d'une concrétisation

Pour des raisons de lisibilité, la traçabilité d'une itération, d'un point de synchronisation et la synchronisation (Globale) ne seront pas représentées. De plus, leurs représentations ne présentent pas un intérêt majeur. La traçabilité est un modèle qui a pour objectif de mener des calculs et interpréter des résultats issus de l'exécution de la synchronisation. Ils n'ont pas vocation à être communiqués.

Chapitre IV CAS D'ETUDE - SYSTEME DE DETECTION ET DE LUTTE INCENDIE

Ce chapitre présente un cas d'étude issu du secteur aéronautique [132]. Il enrichit ainsi l'état des pratiques d'architecture système et d'analyse de la sûreté de fonctionnement.

Ce cas d'étude témoigne de l'utilisation pas à pas des concepts des deux disciplines d'ingénierie concernées. Les études présentées répondent à la formulation d'un besoin client et fournissent une solution d'architecture détaillée représentative d'une partie des activités industrielles. Elles font appel aux concepts liés à la complexité, à l'architecture du système et aux modèles. Pour des raisons d'efficacité de la démonstration, le système a été judicieusement dimensionné.

Le cas d'étude repose sur le respect des standards suivants :

- En ingénierie système : ISO 9000 [106], ISO 15288 [35] ;
- En architecture des systèmes : ISO 42010 [8], IEEE 1220 [33], EIA 632 [34] ;
- En sûreté de fonctionnement ARP 4761 [21], ARP 4754 [131], DO 178-C [133], MIL HDBK 217F [45].

Ce chapitre alimentera le Chapitre V pour illustrer des applications de la synchronisation de modèles en déroulant les étapes de la méthodologie du Chapitre III. Il est construit en trois parties :

- Le système retenu et son contexte d'étude ;
- Les études de conception d'architecture système ;
- Les analyses de sûreté de fonctionnement réalisées.

1. CAS D'ETUDE ET CONTEXTUALISATION

1.1. ORIGINE DU CAS D'ETUDE

APSYS est une société filiale du groupe Airbus spécialisée dans les études de sûreté de fonctionnement et de soutien logistique intégré. Elle développe notamment des outils d'aide à la décision dans ces domaines. Elle a fourni un cas d'étude lors d'un enseignement (« Fiabilité prévisionnelle à l'aide de la MIL-HDBK 217 [45] ») à l'IstiA (Ecole d'ingénieurs de l'Université d'Angers) en 2013. Avec l'accord de la société APSYS, le cas d'étude a été repris et il a été enrichi.

Le système étudié est un détecteur électronique d'incendie intégré dans un hélicoptère de combat. La fonction principale de ce système est de prévenir de l'occurrence d'un feu en le détectant et en avertissant, par des alarmes visuelles et sonores, les occupants de l'hélicoptère. La Figure 84 représente une architecture électrique classique d'un système de détection incendie. Elle est constituée de composants classiques : condensateurs, résistances, détecteur, fusible, lampe, etc.

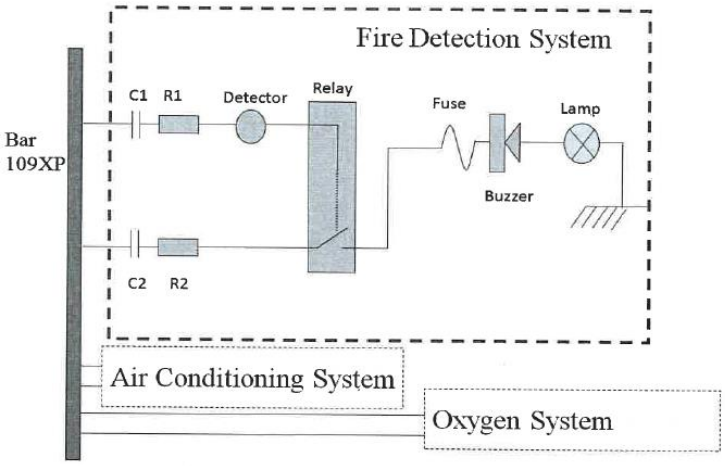


Figure 84 Schéma du système de détection et de lutte contre l'incendie

1.2. CONTEXTE

Pour imaginer un scénario réaliste permettant de contextualiser les études qui suivent, des noms de sociétés fictives sont utilisés : Equipementier A et Systémier B.

1.2.1. L'EQUIPEMENTIER A

L'Equipementier A est spécialisé dans la conception de systèmes embarqués liés aux risques d'incendie. Il dispose de toutes les compétences en interne pour concevoir, vérifier et valider des systèmes sûrs et performants.

1.2.2. LE SYSTEMIER B ET PRESENTATION DU SCENARIO

Le systémier B est fournisseur et concepteur d'hélicoptère. Il développe une nouvelle génération d'hélicoptères militaires pour les armées de certains pays. Ce projet de plusieurs années répond aux besoins des programmes d'armement des dits pays. Les niveaux de performance de ces générations d'hélicoptères sont très sévères et nécessitent d'étudier en profondeur les systèmes.

Une analyse a permis d'identifier les fonctions opérationnelles de l'hélicoptère selon le profil de mission. Plus d'une centaine de fonctions ont été recensées, pour des raisons de lisibilité, la Figure 85 ne représentera que les premières fonctions à l'aide d'un diagramme de cas d'utilisation (UML).

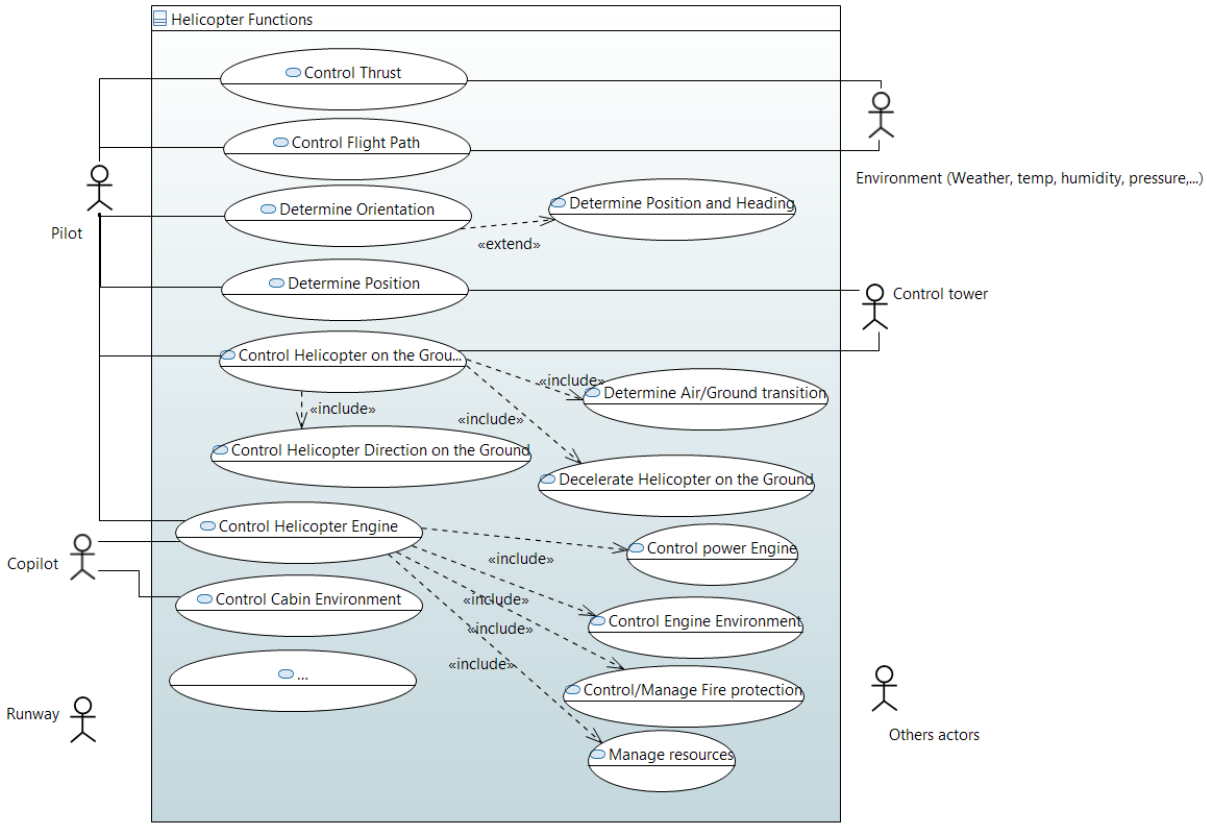


Figure 85 Vue Architecte Système – Analyse opérationnelle de l'hélicoptère

Après des études approfondies sur la motorisation de l'hélicoptère et l'avis d'experts, le systémier B constate un risque incendie de niveau catastrophique (DAL A) dans les zones des moteurs et du rotor (jouant un rôle vitale dans les fonctions de l'hélicoptère).

Ce risque apparait durant l'état de vol en situation de combat. De plus, bien que les moteurs soient performants et résistants aux environnements extrêmes, le blindage des moteurs contre les attaques est un dispositif inenvisageable pour un hélicoptère ayant des propriétés de furtivité et d'agilité au vol. Le systémier B a donc choisi de mettre un dispositif de détection incendie auprès des moteurs et du rotor. Comme elle ne dispose pas des compétences nécessaires, elle a donc ouvert un appel d'offre pour sous-traiter cette activité.

L'Equipementier A a répondu à l'appel d'offre et a été retenu par le systémier B. Le systémier B lui demande alors de concevoir un système de détection et lutte incendie qui doit être embarqué dans son hélicoptère. Il lui fournit un cahier des charges, les contraintes normatives à respecter ainsi qu'une liste d'exigences à vérifier sur le système.

1.3. CAHIER DES CHARGES DU SYSTEME

1.3.1. MISSIONS DU SYSTEME

Le système doit détecter l'occurrence d'un feu dans trois zones spécifiques de l'hélicoptère : celle du moteur principal, celle du moteur secondaire et celle du rotor principal. Ce système automatique d'alarme incendie doit détecter l'apparition de l'évènement redouté « Feu » en surveillant les changements environnementaux associés à la combustion dans les trois zones. Le système doit informer l'équipage et les équipes d'interventions au sol. Le système de détection est alimenté par un bus en 24V courant continue.

D'après le programme de conception de l'hélicoptère, le système sera utilisé dans les conditions environnementales suivantes :

- Température ambiante : 45 °C ;
- Avertissement électrique : 15 °C (valeur relative) ;
- Environnement : hélicoptère.

Pour simplifier le cas d'étude, certaines contraintes ne seront pas présentées comme par exemple l'encombrement ou les contraintes mécaniques bien qu'elles aient de l'importance.

1.3.2. EXPRESSION DU BESOIN

Le systémier B a exprimé son besoin par 7 expressions informelles référencées de N1 à N7 :

- N1 : Besoin d'informer le pilote si les conditions environnementales ne sont pas propices à la mission et source d'incendie au niveau des zones surveillées ;
- N2 : Besoin d'atterrissage forcé en cas d'incendie non contrôlable et besoin d'une détection très rapide ;
- N3 : Besoin d'une détection dans le cas d'évènements redoutés de type incendie ;
- N4 : Besoin d'une première intervention au départ du feu ;
- N5 : Besoin d'informer les occupants par un signal en cas d'incendie ;
- N6 : Besoin d'informer une équipe au sol de la situation pour une éventuelle évacuation ;
- N7 : Besoin d'une source d'alimentation d'urgence en cas d'incident.

A partir des échanges avec le client, une première traduction du besoin en exigences techniques de spécification du système a été établie. Elles sont référencées par SysReq001 à SysReq003 (System Requirement) et ReqFunct001 à ReqFunct004 (Functional Requirement). La liste présentée Table 14 n'est pas exhaustive.

Table 14 Spécification des exigences

ID *l'exigence*	***Description***
SysReq001	*Le système de protection doit avoir un taux de défaillance d'au moins 10^{-5} h^{-1}*
SysReq002	*Le système de lutte doit avoir un taux de défaillance d'au moins 10^{-5} h^{-1}*
SysReq003	*Le temps de maintenance ne dépassera pas 1h au sol en cas d'utilisation. Une maintenance préventive est prévue 1 fois par mois*
ReqFunct001	*Le système de détection et de lutte incendie est en charge de surveiller le départ de feu, d'alerter les occupants et les acteurs hors site en cas de feu et de réaliser une première lutte incendie sur les zones sans les détériorer pour permettre une évacuation.*
ReqFunct002	*Le système doit surveiller et détecter le départ de feu. Il observera dans toutes les conditions de vol le départ d'un feu dans la zone surveillée. Cette détection permettra d'envoyer un signal aux acteurs concernés.*
ReqFunct003	*Le système doit lutter contre l'incendie. Il faut prévoir un dispositif d'activation de lutte commandé par un des occupants de l'hélicoptère. En cas de non réponse, l'hélicoptère activera la lutte dans un délai de 2min.*
ReqFunct004	*Le système doit alerter en cas d'incendie les occupants de l'hélicoptère et une équipe hors site (un aéroport à proximité par exemple) qui pourra prévoir une équipe d'intervention d'urgence. Le message détaillera les caractéristiques du vol, de la mission et du départ de feu.*

2. PRESENTATION DES TRAVAUX DE L'ARCHITECTURE SYSTEME SUR LE CAS D'ETUDE

2.1. LE PROCESSUS DES ACTIVITES D'ARCHITECTURE SYSTEME

Pour mener les activités d'architecture sur le système étudié, en appliquant une démarche basée sur les modèles, l'outil Papyrus [134] a été utilisé pour construire des modèles SysML [2]. Cet outil permet de construire des vues du système à différents niveaux de raffinement et d'avancement dans le projet.

Les activités demandées par les normes ont été déclinées sur le processus métier d'architecte système chez l'Equipementier A. Ce processus se décline en 3 étapes successives (cf. Figure 86) : l'analyse opérationnelle, la conception fonctionnelle et la conception physique.

Ces trois étapes sont menées de manière incrémentale et comprennent chacune une étape systématique de vérification et de communication avec les parties prenantes. Le diagramme BPMN [27] montre l'ordonnancement logique des activités utilisées dans le cas d'étude.

Toutes les activités, méthodes et points de vue sont construits et capitalisés dans des modèles SysML. Les éléments, les propriétés et les diagrammes sont contenus dans des packages. Ces derniers sont structurés de sorte à refléter le processus mené.

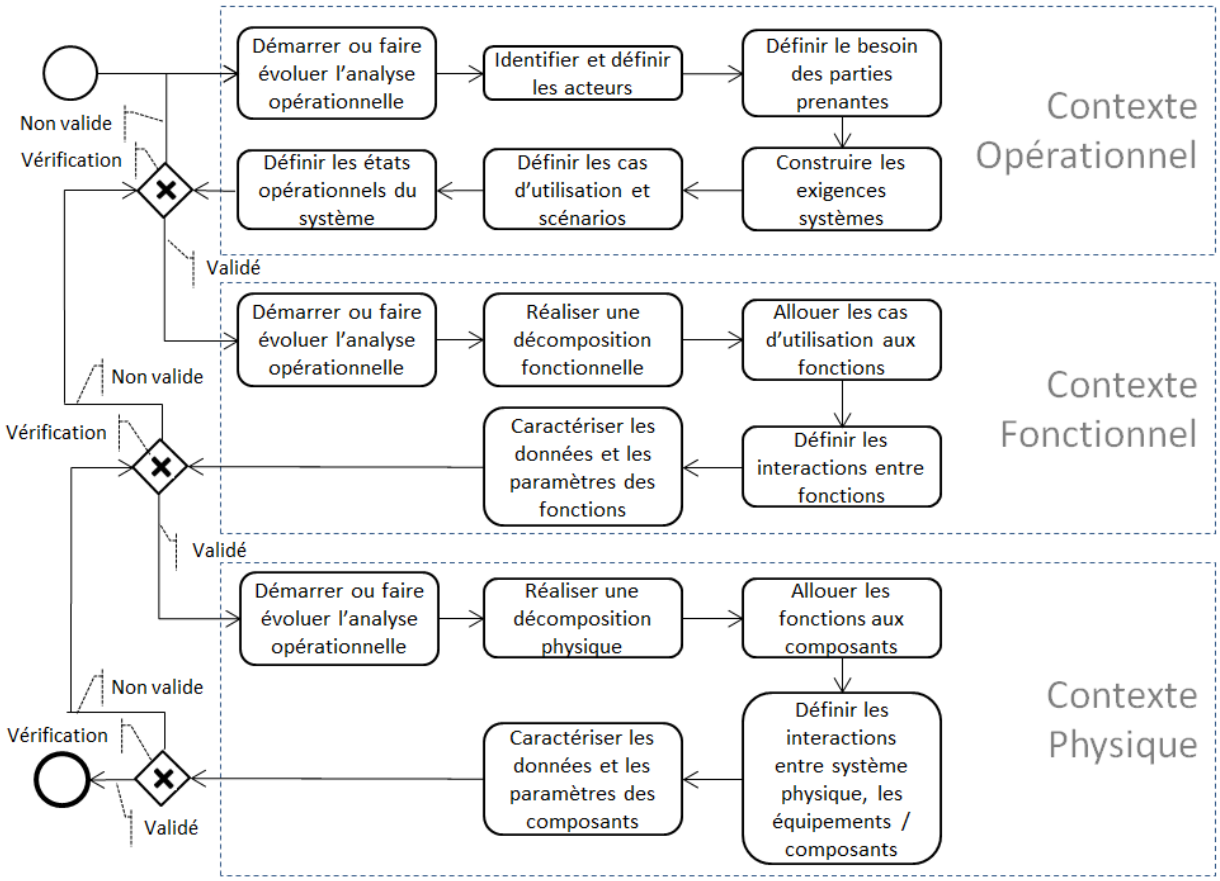


Figure 86 Diagramme BPMN du processus de l'Equipementier A de l'architecture système

2.2. ANALYSE OPERATIONNELLE

Les parties prenantes du système de détection et de lutte incendie sont listées dans la Table 15.

Table 15 Parties prenantes du système

Acteur du système	**Description**	**Contraintes et spécificités**
Acteur hors site organisme de secours	Il s'agit d'une zone externe à l'hélicoptère. Cette zone hors site pourra recevoir des messages d'alerte en provenance de l'hélicoptère.	Les messages transmis aux acteurs hors site pourront être de différents types : message d'information sur l'état de l'hélicoptère, message d'information relative à la mission, message d'information relative à une demande d'intervention.
Occupants de l'hélicoptère	Deux occupants sont en charge du pilotage de l'hélicoptère. Placés à l'avant de l'hélicoptère, leur zone de vie est isolée physiquement et hermétiquement.	Pilotes entrainés, occupants en pleines conditions physique et mentale pour le pilotage.
Les moteurs et le rotor	Le système sera partiellement placé dans les compartiments moteurs et du rotor, en situation de vol.	- Encombrement disponible 0,8 10^{-3} m^3. - Température autour du moteur : 20 à 150 °C, en phase de vol. - Bruit : 130 Décibel (http://www.industrialnoisecontrol.com/comparative-noise-examples.htm) - Vibration de 2 à 5 GRMS - Pression de 500 à 1500 HPa - Altitude de -420 à 5000 m - Humidité identique par rapport à l'extérieur - Poids total autorisé pour les dispositifs de détection et de lutte incendie 5kg
L'hélicoptère de combat	C'est le support d'installation où reposera les composants du système.	Contraintes cabine liées aux dispositifs d'alerte : - Les composants non relatifs à la détection et à la lutte incendie devront être positionnés dans l'habillage de la cabine. - Encombrement disponible 0.001m^3 - Poids total autorisé pour les dispositifs est de 5kg
Environnement extérieur	C'est l'environnement nominal en situation de vol dans une zone géographique déterminée. Il représente l'environnement externe de l'hélicoptère.	- Température ambiante -12 à 60 °C - Vibration de 0 à 4 GRMS - Pression de 500 à 1500 HPa - Altitude de -420 à 5000m - Humidité de 40 à 97%
Energie électrique	L'alimentation électrique provient de deux batteries « Saft » redondantes.	- Alimentation principale : La batterie 275CH3 comprend 20 éléments SAFT CVH27 fournissant une tension nominale de 24Vcc et une capacité de 27Ah pour un poids total de 26,5kg - Alimentation de secours : Un réseau de secours alimenté en 12V et 50Ah

A partir de la liste des besoins du client et de la définition des parties prenantes, les besoins ont été alloués aux acteurs, cf. Table 16 et Figure 87.

Table 16 Allocation des besoins aux acteurs

Acteurs \ Besoins	N1	N2	N3	N4	N5	N6	N7
Acteur hors site organisme de secours		X			X		
Occupants de l'hélicoptère						X	
Les moteurs et le rotor			X	X			
L'hélicoptère de combat		X					
Environnement extérieur	X						
Energie électrique							X

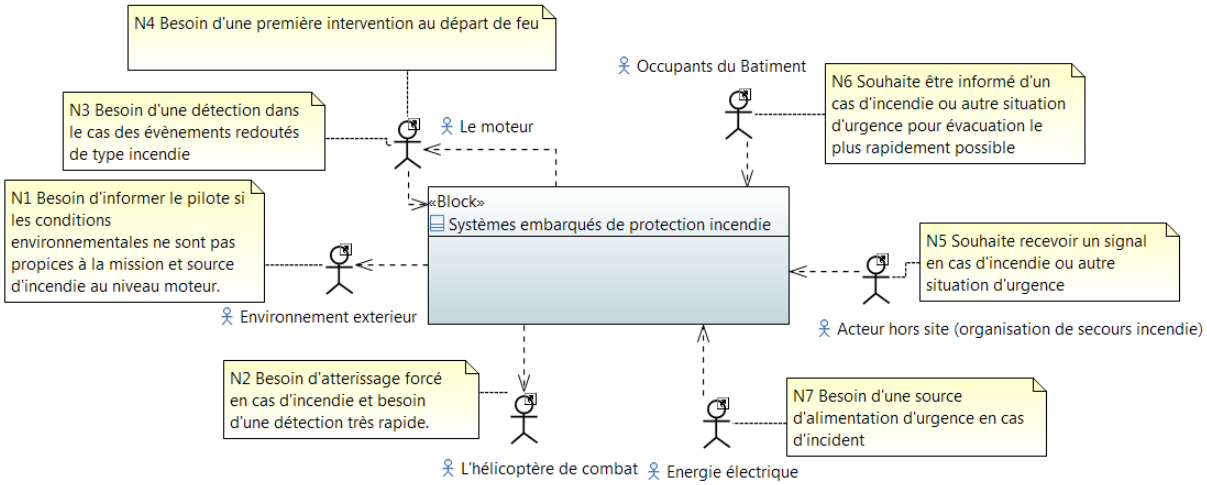


Figure 87 Allocation des besoins aux acteurs

Un diagramme de cas d'utilisation du système a été construit pour expliciter les usages du système par les parties prenantes et les systèmes externes. Trois cas d'utilisation sont identifiés : surveiller et détecter le départ d'un feu, envoyer une alerte, enfin lutter contre l'incendie. Le diagramme de cas d'utilisation (UML) est présenté dans la Figure 88 et commenté dans la Table 17.

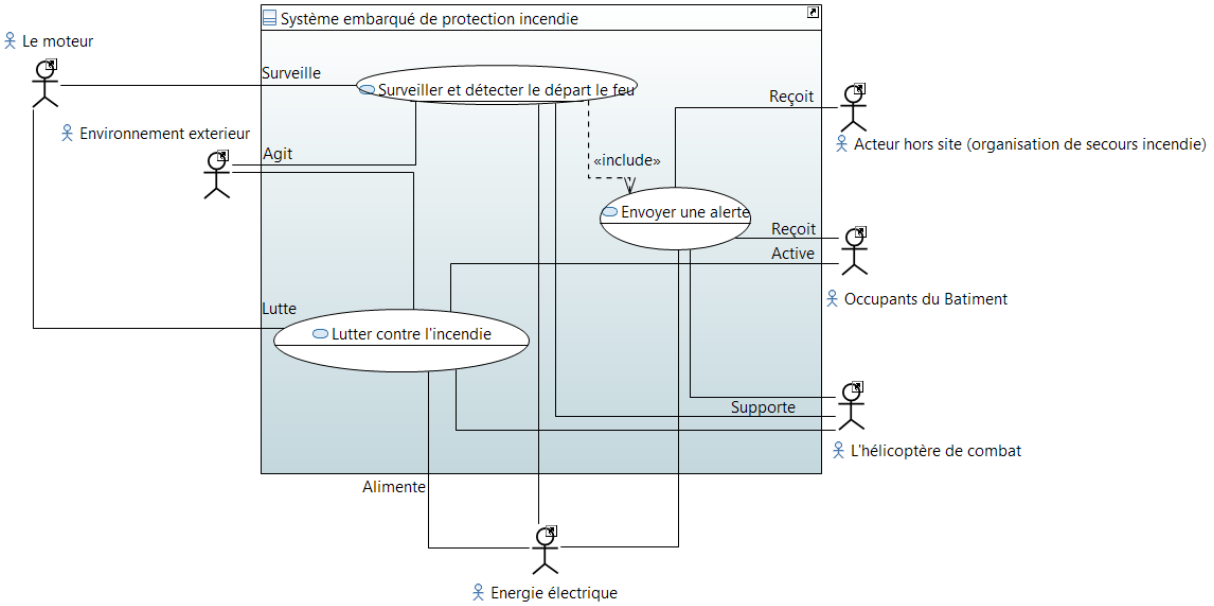


Figure 88 Diagramme UML de cas d'utilisation du système

Table 17 Description des cas d'utlisation

<u>Cas d'utilisation n°1 : Surveiller et détecter le départ de feu</u> *Version : 1.2 ;* *Description : Mesure et détecte l'occurrence d'un feu dans une zone surveillée* *Acteurs : Moteurs et rotor, Environnement extérieur, Hélicoptère de combat, Energie électrique.* *Référence : Besoins client, Cahier des charges et Spécifications client* *Prérequis : L'hélicoptère est en vol. Un feu se déclenche dans l'une des trois zones surveillées.* *Conséquences : Suite à la détection du feu, un signal est envoyé.* *Scénario :* *- Un des environnements des moteurs ou du rotor évolue suite au départ d'un feu.* *- Des capteurs du système recueillent des mesures sur un des environnements des moteurs ou du rotor. Ils identifient la présence d'un feu.*

- Envoi d'alerte à travers le cas d'utilisation « Envoyer une alerte ». *Exception :* *- Le feu (ou la cause du feu) a détruit le système avant sa détection*
<u>Cas d'utilisation n°2 : Envoyer une alerte</u> *Version : 1.1 ;* *Description : Signaler aux occupants et à une équipe au sol la détection d'un feu* *Acteurs : Acteur hors-site, Occupants de l'hélicoptère, Energie électrique* *Référence : Besoins client, Cahier des charges et Spécifications client* *Prérequis : L'hélicoptère est en vol. Un feu se déclenche dans l'une des trois zones surveillées. Celui-ci a été détecté.* *Conséquences : Les occupants et l'acteur hors-site sont informés de l'incendie.* *Scénario :* *- Le système d'alerte reçoit un signal de détection d'un feu.* *- Déclenchement des signaux d'alerte (visuels et sonores) dans la cabine* *- Envoi d'un signal à distance pour informer l'acteur hors-site* *<u>Pas d'exception.</u>*
Cas d'utilisation n°3 : Lutter contre l'incendie *Version : 2.0 ;* *Description : Gérer le déclenchement de la lutte incendie* *Acteurs : Hélicoptère de combat, Occupants de l'hélicoptère, Energie électrique, Environnement extérieur, Moteurs et rotor* *Référence : Besoins client, Cahier des charges et Spécifications client* *Prérequis : L'hélicoptère est en vol. Un feu se déclenche dans l'une des trois zones surveillées. Celui-ci a été signalé.* *Conséquences : Le feu est éteint* *Scénario :* *- Demande auprès des occupants si la lutte incendie doit être lancée (demande d'un délai de 2min)* *- Les occupants valident la lutte incendie ou les occupants n'ont pas donné de refus à temps.* *- Déclenchement de la lutte incendie dans la zone surveillée.* *Exceptions :* *- Le feu est trop puissant et la lutte incendie n'a pas été efficace*

2.3. SPECIFICATION DES EXIGENCES TECHNIQUES

A partir des spécifications de haut-niveau du système et de l'analyse opérationnelle, il a été mené une déclinaison des exigences référencées sur les cas d'utilisations UC dans la Table 18. Ces exigences devront être validées en testant le système sur 1000 heures (MT0) avec une durée moyenne d'une heure par vol (FT0).

Table 18 Déclinaison des exigences aux cas d'utilisation

ID *exigence*	***Cas d'utilisation correspondant***	**Partie prenantes concernées**	***Description***
R026	*Surveiller et détecter le départ de feu*	*Moteurs & rotor*	Le système doit détecter un feu en moins de 3 secondes après la détection du feu.
R027	*Surveiller et détecter le départ de feu*	*Moteurs & rotor*	Le système ne doit pas perdre la détection d'incendie avec un taux de défaillance inférieur à 10^{-6} h^{-1}.
R028	*Surveiller et détecter le départ de feu*	*Moteurs & rotor*	Le système ne doit pas alerter, ni détecter d'incendie dans le cas contraire (faux-vrai), avec un taux de défaillance inférieur à 10^{-4} h^{-1}.
R029	*Surveiller et détecter le départ de feu*	*Moteurs & rotor*	La détection doit permettre de localiser l'incendie (précision demandée à 50 cm près).

ID *exigence*	***Cas d'utilisation correspondant***	**Partie prenantes concernées**	***Description***
R030	*Surveiller et détecter le départ de feu*	*Moteurs & rotor*	Le taux de défaillance (MTBF) du système de détection incendie doit être inférieur à 10^{-4} h^{-1} (10 000 heure)
R031	*Lutter contre un incendie*	*Moteurs & rotor*	Le dispositif ne devra pas endommager le moteur. La technologie employée devra permettre d'étouffer les flammes sans apporter de poussière ou d'impureté dans la zone.
R032	*Surveiller et détecter le départ de feu*	*Energie électrique*	L'alimentation principale se fera par 2 batteries Saft, 24 Vcc et 27 Ah. Elle alimentera l'ensemble des composants de détection.
R033	*Surveiller et détecter le départ de feu*	*Energie électrique*	Alimentation de secours : Un réseau de secours alimenté en 12 Vcc et 50 Ah. Ce réseau ne sera utilisé qu'en cas de perte énergétique au niveau de l'alimentation principale. Elle a pour objectif d'alimenter les composants vitaux de l'hélicoptère. Ces derniers seront soumis à une évaluation sur des scénarios d'accidents tests. Les composants doivent répondre à un besoin sur la mission d'atterrissage d'urgence ou de détection à risques tels que les risques d'incendie.
R034	*Surveiller et détecter le départ de feu*	Hélicoptère de combat (+ Autorité)	En cas de conditions environnementales non acceptables, l'hélicoptère ne sera pas autorisé à décoller.
R035	*Surveiller et détecter le départ de feu*	Hélicoptère de combat	Encombrement disponible 0,8 m^3 au niveau moteur partagé avec le système de lutte incendie
R037	*Surveiller et détecter le départ de feu*	Hélicoptère de combat	Une IHM est prévue pour les communications entre le pilote et le système. La surface est de 150 cm^2. Alimentée en 24V par le réseau énergétique de l'hélicoptère, l'interface devra être reliée au circuit de secours.
R036	*Surveiller et détecter le départ de feu*	Hélicoptère de combat	Poids total autorisé pour les dispositifs est de 5kg partagé avec le système de lutte incendie.
R016	*Surveiller et détecter le départ de feu*	Hélicoptère de combat	Encombrement disponible 0,33m^3
R017	*Surveiller et détecter le départ de feu*	Hélicoptère de combat	Le poids total autorisé pour les dispositifs est de 4,2kg.
R018	*Surveiller et détecter le départ de feu*	Hélicoptère de combat	Cf R037
R019	*Surveiller et détecter le départ de feu*	Energie électrique	L'alimentation principale se fera par 2 batteries Saft, 24Vcc et 27Ah. Elle alimentera l'ensemble des composants du système.
R020	*Surveiller et détecter le départ de feu*	Energie électrique	Cf R033
R021	*Surveiller et détecter le départ de feu*	Moteurs & rotor	Encombrement disponible 0,40m^3
R023	*Surveiller et détecter le départ de feu*	Moteurs & rotor	Température autour du moteur : 20 à 150 °C

ID *exigence*	***Cas d'utilisation correspondant***	**Partie prenantes concernées**	***Description***
R024	*Surveiller et détecter le départ de feu*	Environnement extérieur	Bruit : 130 dB (http://www.industrialnoisecontrol.com/comparative-noise-examples.htm) (intervalle de confiance)
R025	*Surveiller et détecter le départ de feu*	Occupants de l'hélicoptère	Vibration de 2 à 5 GRMS
R038	*Surveiller et détecter le départ de feu*	Occupants de l'hélicoptère	Pression de 500 à 1500 HPa
R001	*Envoyer une alerte*	Occupants de l'hélicoptère	Délai d'alerte après détection ingénieur ou égal à 1sec
R002	Envoyer une alerte	Occupants de l'hélicoptère	L'installation en cabine devra disposer de 2 types de signalisation : 1 sonore, 1 visuelle
R003	Envoyer une alerte	Occupants de l'hélicoptère	Le système doit envoyer un message à chaque détection, avec un taux d'échec inférieur à 10^{-6} h^{-1}.
R005	Envoyer une alerte	Occupants de l'hélicoptère	Le système ne doit pas envoyer de signal si aucun feu n'est détecté, avec un taux d'échec inférieur à 10^{-4} h^{-1}.
R006	Envoyer une alerte	Organismes de secours incendie	Délai entre l'envoi de l'alerte et réception par l'autorité au sol inférieur ou égal à 1min
R009	Envoyer une alerte	Organismes de secours incendie	Le système ne doit pas envoyer de signal si aucun feu n'est détecté, avec un taux d'échec inférieur à 10^{-6} h^{-1}.
R010	Envoyer une alerte	Organismes de secours incendie	L'envoi d'une alerte devra être sécurisé. Un message crypté est transmis par le système "NETEM" par satellite. Ce système réorientera le message directement à l'autorité sur place au sol. Le périmètre de conception s'arrête à ce niveau.
R011	Envoyer une alerte	Hélicoptère de combat	Encombrement disponible 0,33m^3
R012	Envoyer une alerte	Hélicoptère de combat	Cf R017.
R013	Envoyer une alerte	Hélicoptère de combat	Un emplacement pour une IHM est prévu pour les communications entre le pilote et le système. La surface est de 150 cm^2. Alimenté en 24V par le réseau énergétique de l'hélicoptère. Il devra être relié au circuit de secours.
R014	Envoyer une alerte	Energie électrique	Cf R019
R015	Envoyer une alerte	Energie électrique	Cf R033

2.4. DEFINITION DE L'ARCHITECTURE FONCTIONNELLE

2.4.1. DECOMPOSITION FONCTIONNELLE

L'analyse opérationnelle permet maintenant de construire un découpage fonctionnel du système. La fonction principale du système est FC1 « Lutter contre les incendies », elle se décompose ainsi :

- FC11 : Détecter un départ de feu moteur ;
  - FC111 : Mesurer le taux de monoxyde de carbone ;
  - FC112 : Mesurer la présence de fumée ;
  - FC112 : Interpréter les mesures en fonction des données d'entrées ;

- FC12 : Lutter contre un départ de feu moteur ;
  - FC121 : Activer un système de lutte incendie ;
- FC13 : Informer d'un incendie moteur ;
  - FC131 : Alerter les occupants de l'hélicoptère par un message sonore et visuel ;
  - FC132 : Alerter les acteurs hors site ;
- FC14 : Assurer une alimentation de 12Vcc ;
  - FC141 : Transformer le courant d'entrée de 24Vcc à 12Vcc ;
  - FC142 : Distribuer le courant aux composants au niveau moteurs et cabine ;
  - FC143 : Relier à la tension de secours en cas de non alimentation nominale (Redondance de la fonction d'alimentation à l'extérieur du système).

Cette décomposition permet de construire le diagramme de définition des Blocks de la Figure 89 :

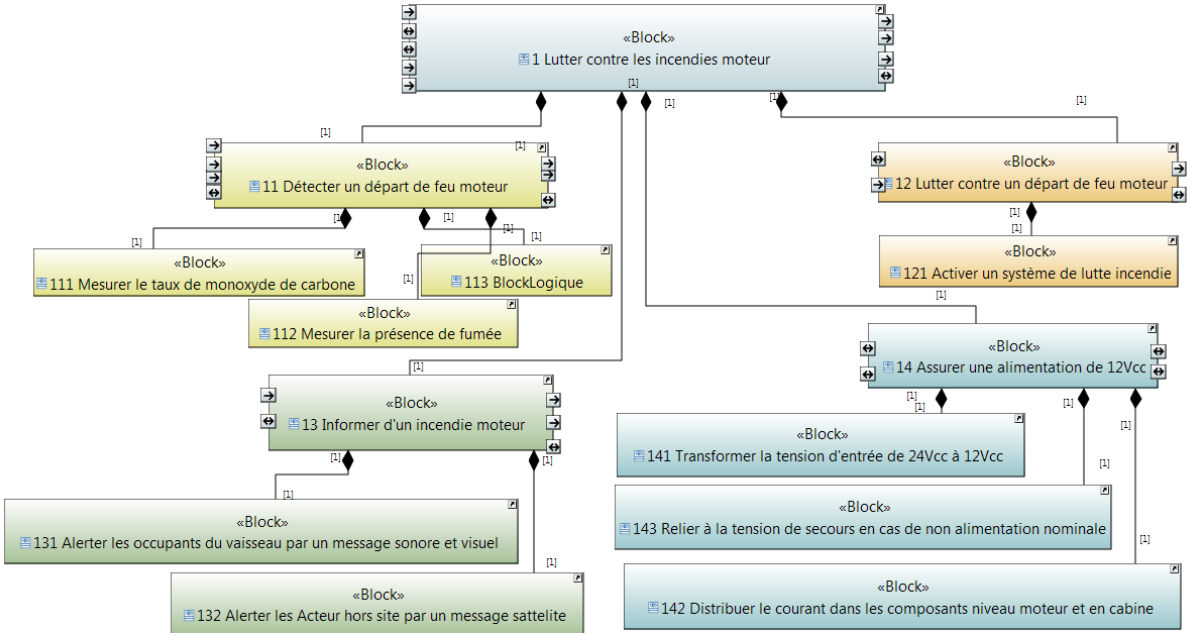


Figure 89 Décomposition fonctionnelle du système par un diagramme de définition des blocks SysML

2.4.2. ALLOCATION DES FONCTIONS AUX CAS D'UTILISATION

Les cas d'utilisation sont ensuite alloués aux fonctions définies précédemment. Les sous-fonctions sont allouées de la même manière que leurs fonctions parentes.

Table 19 Allocation des fonctions aux cas d'utilisation

Cas d'utilisation /Fonction	FC1	FC11	FC12	FC13	FC14
UC 1 : *Surveiller et détecter le départ de feu*	X	X			X
UC 2 : *Envoyer une alerte*	X			X	X
UC 3 : *Lutter contre l'incendie*	X		X		X

2.4.3. INTERCONNEXIONS DES FONCTIONS

Pour l'assemblage des fonctions, les interactions sont définies et représentées par un diagramme de block interne (SysML). L'opération a été déclinée sur toutes les fonctions ayant des sous-fonctions.

A ce niveau d'étude, les représentations fonctionnelles ne discriminent pas les zones surveillées. Elles sont abstraites et seront instanciées dans l'architecture physique.

Les figures suivantes (Figure 90, Figure 91, Figure 92, Figure 93 et Figure 94) présentent les décompositions des fonctions FC par des diagrammes de block interne.

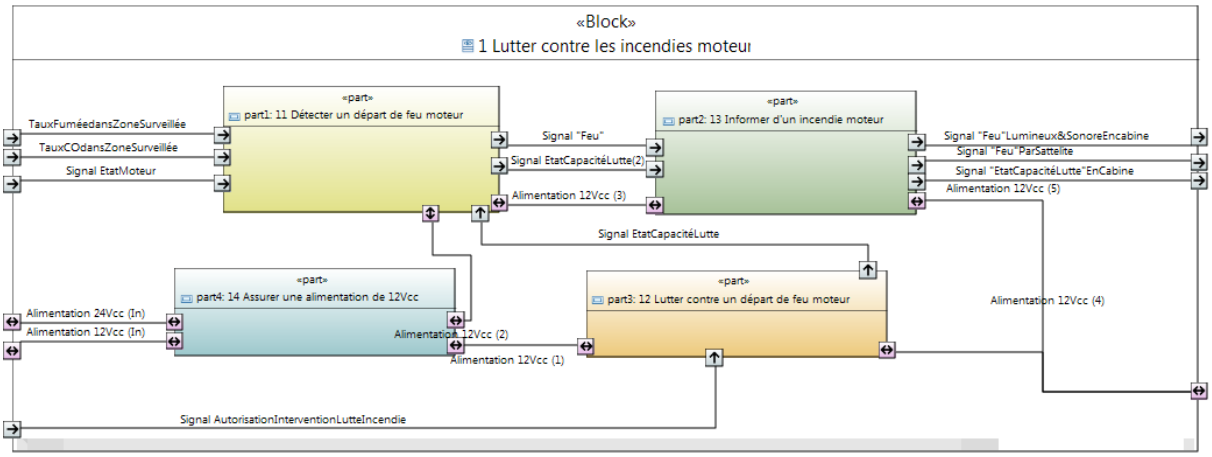


Figure 90 Diagramme de block interne de la fonction 1 : Lutter contre les incendies moteurs

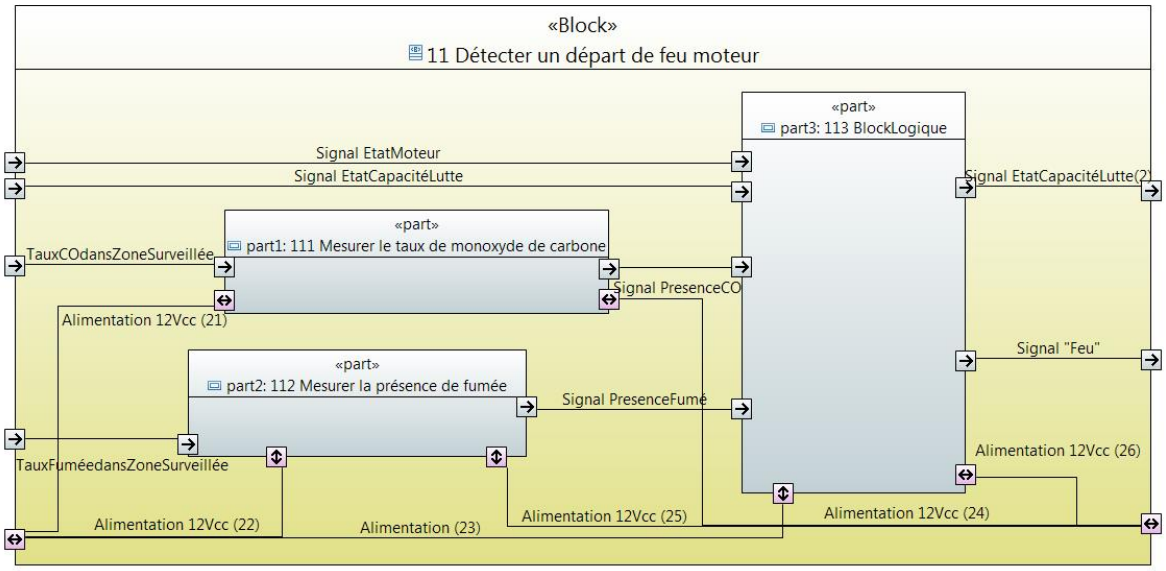


Figure 91 Diagramme de block interne de la fonction 11 : Détecter un départ de feu

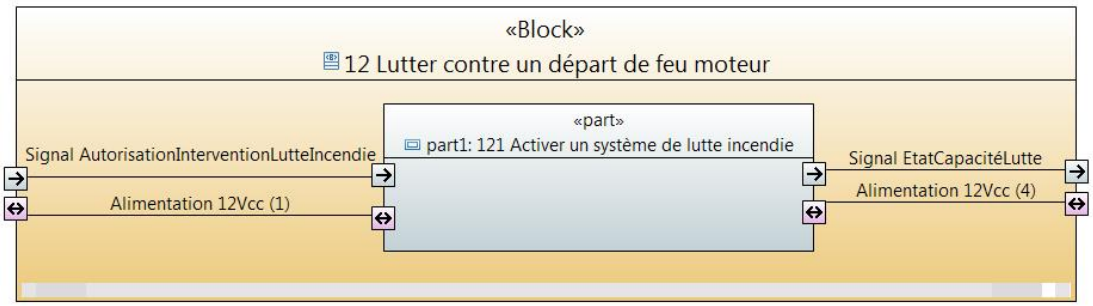


Figure 92 Diagramme de block interne de la fonction 12 : Lutter contre un départ de feu moteur

La conception du sous-système répondant à la fonction « Lutter contre un départ de feu moteur » a été soutraitée à une société externe. Elle devra réaliser la conception d'un système en respectant les spécifications. Elle devra également garantir de la sûreté de sa solution.

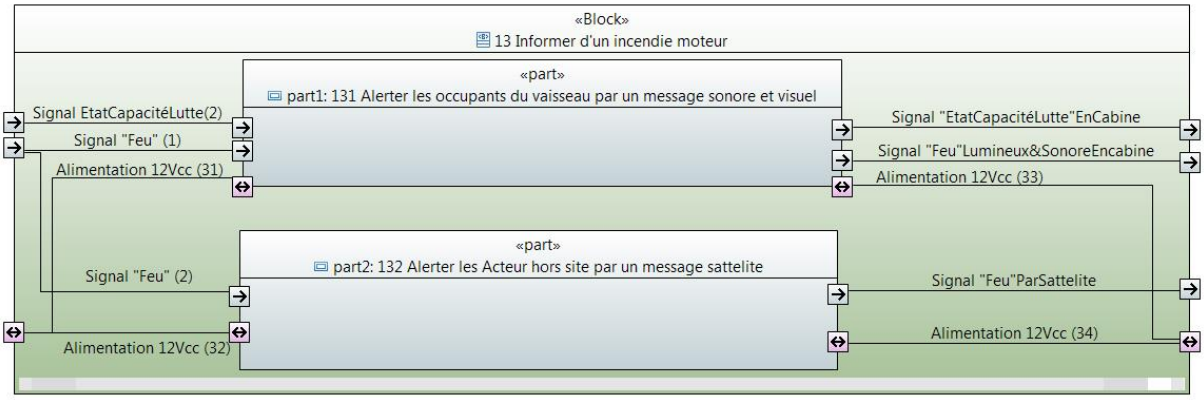


Figure 93 Diagramme de block interne de la fonction 13 : Informer d'un incendie moteur

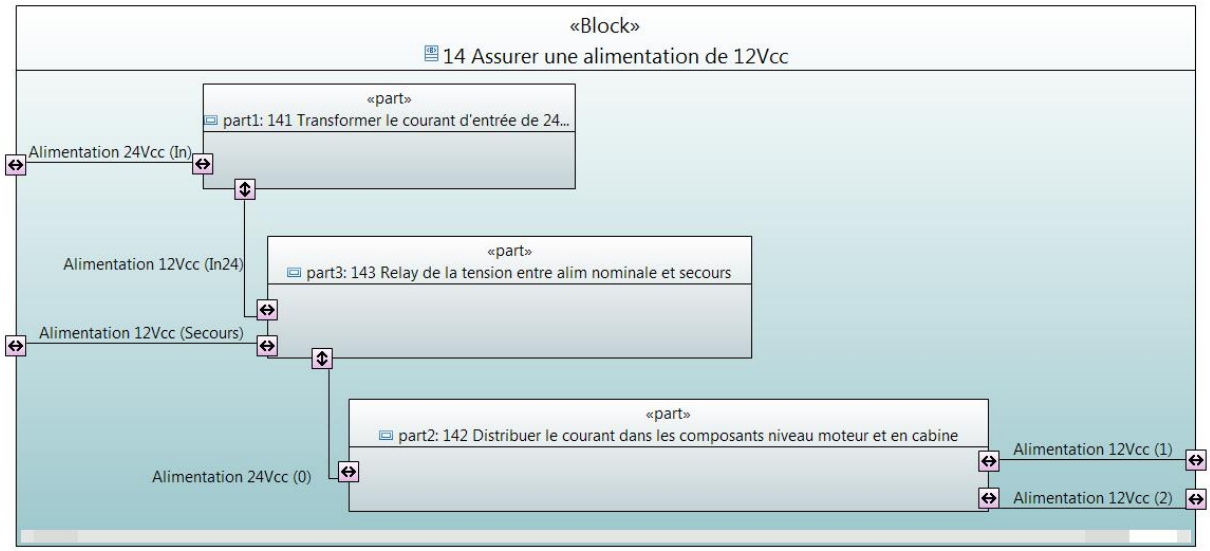


Figure 94 Diagramme de block interne de la fonction 14 : Assurer une alimentation de 12 Vcc

2.5. DEFINITION D'UNE ARCHITECTURE PHYSIQUE

2.5.1. DECOMPOSITION AU NIVEAU DES COMPOSANTS PHYSIQUES

A partir de l'architecture fonctionnelle du système, un découpage physique du système est réalisé. Le système de détection et lutte incendie, Sys1, a pour sous-systèmes :

- Sys11- 3 Ensemble Capteur Feu ;
  - C111 - Capteur de monoxyde de carbone ;
  - C112 - Capteur de fumée ;
  - C113 - Récepteur signal « Lutte achevée » ;
  - C114 - Bloc de contrôle logique de mesure ;
  - C115 - Emetteur signal « Alerte Feu » ;
- Sys12- 1 Dispositif de lutte ;
  - C121 - Récepteur signal « Activation Lutte » ;
  - C122 – Actionneur ;
  - C123 - Liquide d'extinction + Conteneur ;
  - C124 - Emetteur Signal « Lutte achevée » ;
- Sys13- 1 Dispositif d'alerte ;
  - C131 - 10 LEDs rouges en série

- C132 – Buzzer ;
- C133 - Emetteur radio de signal ;
- C134 – Fusible ;

- Sys14- 1 Réseau d'alimentation ;
  - C141 – Convertisseur ;
  - C142 – Résistance ;
  - C143 – Relay ;
  - C144 - Fils électrique ;
  - C145 – Condensateur.

La Figure 95 illustre la décomposition des fonctions par un diagramme de définition de block en SysML.

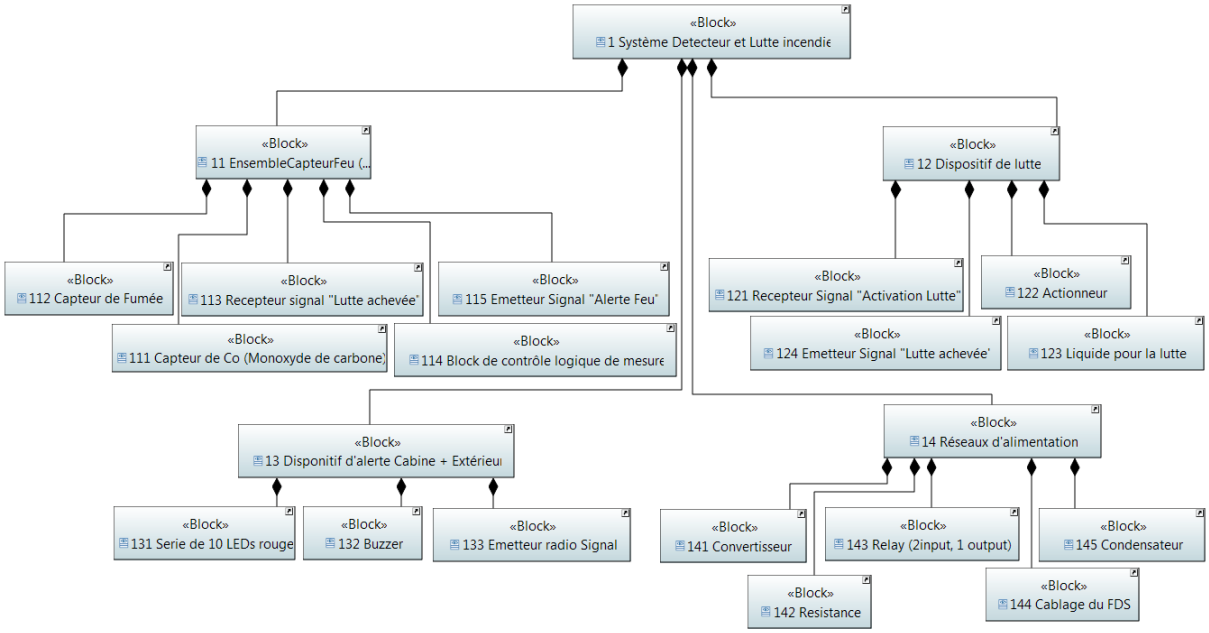


Figure 95 Décomposition de l'architecture physique

2.5.2. ALLOCATION DES FONCTIONS ET COMPOSANTS

De la même manière que pour les cas d'utilisation UC et fonction FC, il convient d'allouer les fonctions FC aux composants Sys et C contenus dans la décomposition physique. Les fonctions sont remplies par les composants alloués Table 20.

Table 20 Allocation des fonctions aux composants

Fonction/composant	Sys1	Ssys11	Ssys12	Ssys13	Ss14	C111	C112	C113	C114	C115	C121	C122	C123	C124	C131	C132	C133	C141	C142	C143	C144	C145
FC1- Lutter contre les incendie moteurs	x																					
FC11 - Detecter un départ de feu moteur		x																				
FC12 - Lutter contre un départ de feu moteur			x																			
FC13 - Informer d'un incendie moteur				x																		
FC14 - Assurer une alimentation de 12Vcc					x																	
FC111 - Mesurer le taux de monoxyde de carbone						x																
FC112 - Mesurer la présence de fumée							x															
FC113 - BlockLogique, Interpréter les mesures									x													
FC121- Activer un système de lutte incendie											x	x	x	x								
FC131 - Alerter les occupants du vaisseau par un message sonore et visuel										x					x	x			x	x	x	x
FC132 - Alerter les acteurs hors site										x							x		x	x	x	x
FC141- Transformer la tension d'entrée de 24Vcc à 12Vcc																		x				
FC142 - Distribuer le courant dans les composants au niveau des moteurs et en cabine																					x	
FC143 - Relayer la tension de secours en cas de non alimentation nominale	Non considéré dans la suite de l'étude (délégué à la fonction d'alimentation à l'extérieur du système)																					

2.5.3. INTERCONNEXIONS DES COMPOSANTS

Les composants et les sous-systèmes ont ensuite été assemblés entre eux. Les interactions sont définies et représentées dans les diagrammes de blocks internes sur l'ensemble du système physique (cf. Figure 96, Figure 97, Figure 98, Figure 99 et Figure 100).

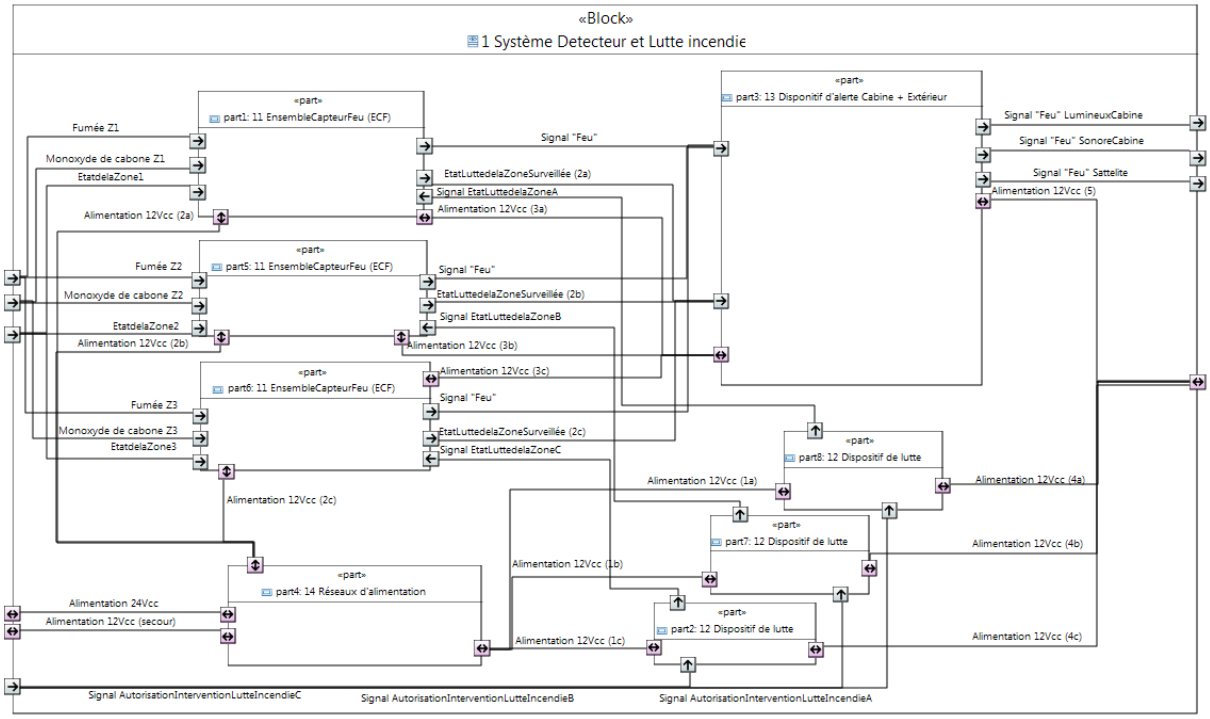


Figure 96 Interactions des sous-systèmes dans un diagramme de blocks internes

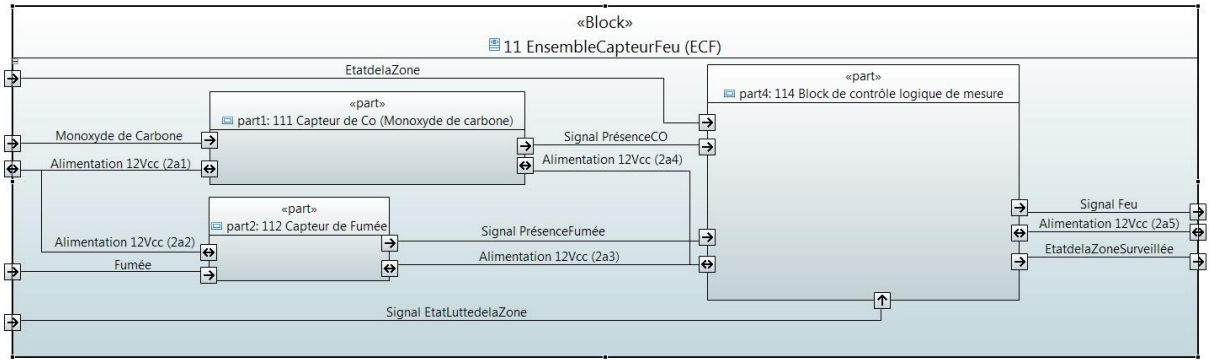


Figure 97 Interactions des composants du sous-système Ensemble Capteur de feu (ECF)

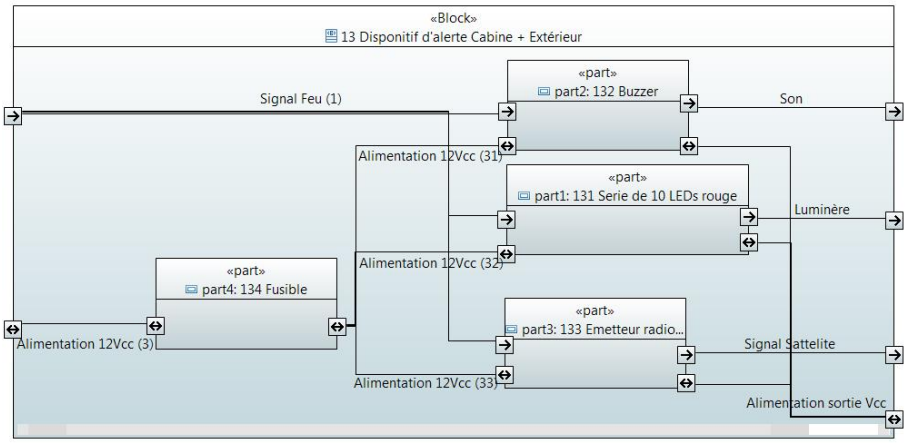


Figure 98 Interactions des composants du sous-système d'alerte cabine et extérieur

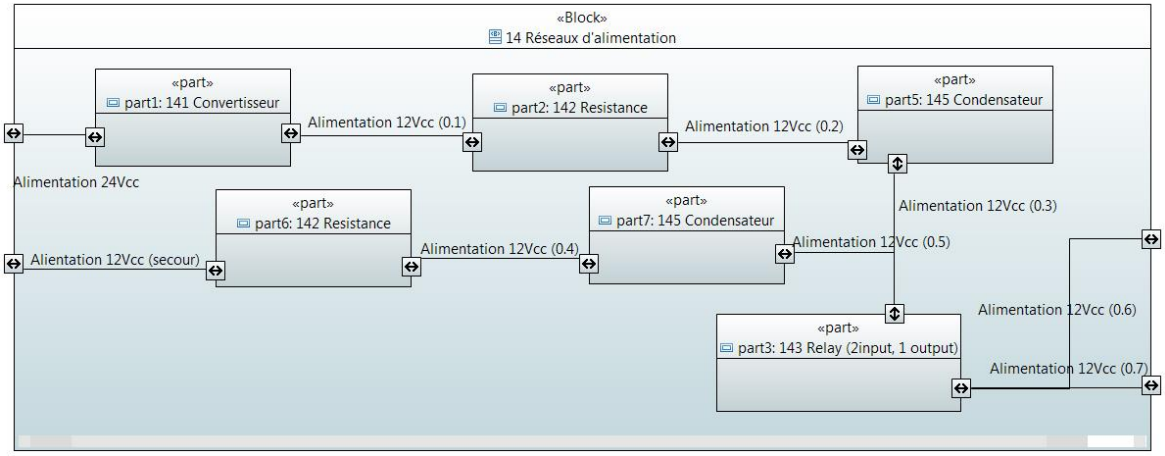


Figure 99 Interaction des composants du sous-système réseaux d'alimentation

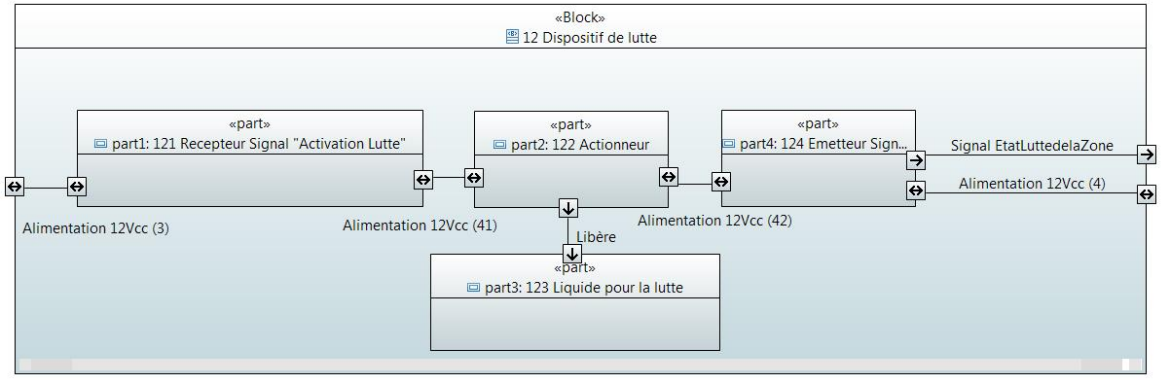


Figure 100 Interaction des composants du sous-système dispositif de lutte

3. PRESENTATION DES TRAVAUX DE L'INGENIEUR EN SURETE DE FONCTIONNEMENT SUR LE CAS D'ETUDE

Une démarche basée sur les modèles (MBSA) a été adoptée pour mener les activités de sûreté de fonctionnement sur le système étudié. Les outils suivants ont été utilisés : un tableur, l'atelier AltaRica [70], [71] ainsi que les outils exploitants les fichiers OpenPSA [135] (Arbre Analyste). Celui-ci permet de construire des vues spécifiques du métier à différents niveaux de raffinement.

Les activités issues des normes aéronautiques (cf. Figure 101) ont été déclinées en un processus métier de la sûreté de fonctionnement chez l'Equipementier A. Il respecte les recommandations de l'ARP 4754 [131], la DO 178-C [133] et utilise les guides MIL HDBK-217 [45] et UTE C80 810 [49] pour l'estimation et l'allocation de la fiabilité.

Le processus se veut itératif, c'est pourquoi les étapes de vérification et d'interaction avec les parties prenantes sont menées systématiquement à la suite des activités.

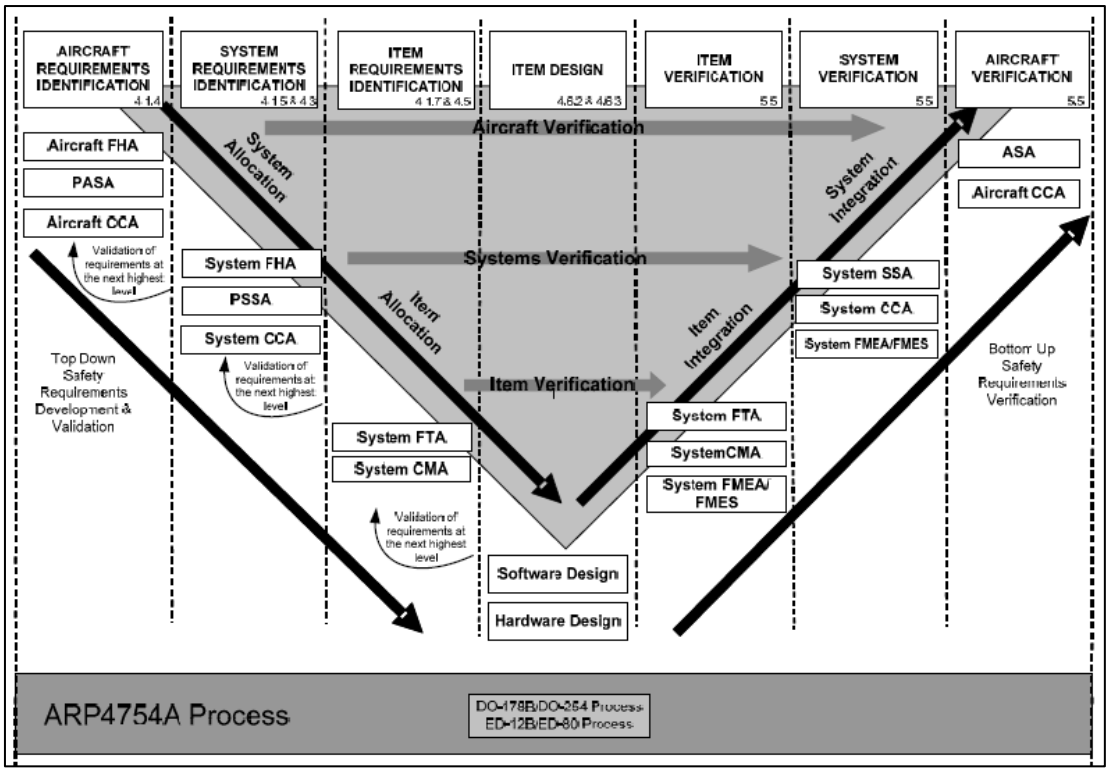


Figure 101 Extrait du processus de l'ARP 4754

Les activités encadrées en rouge, dans la Figure 102, sont présentées dans ce chapitre :

- Un extrait de l'Aircraft FHA, PSSA, System FHA, PASA pour la phase descendante du cycle de développement ;
- Une étude PSA pour une partie de la phase remontante du cycle de développement. L'étude PSA (probabilistic safety assessment) a permis de construire les AMDECs, les FTAs et les MCS Minimal CutSet (les coupes minimales). Les coupes minimales sont l'ensemble des évènements de base qui cumulé amène directement à l'évènement redouté.

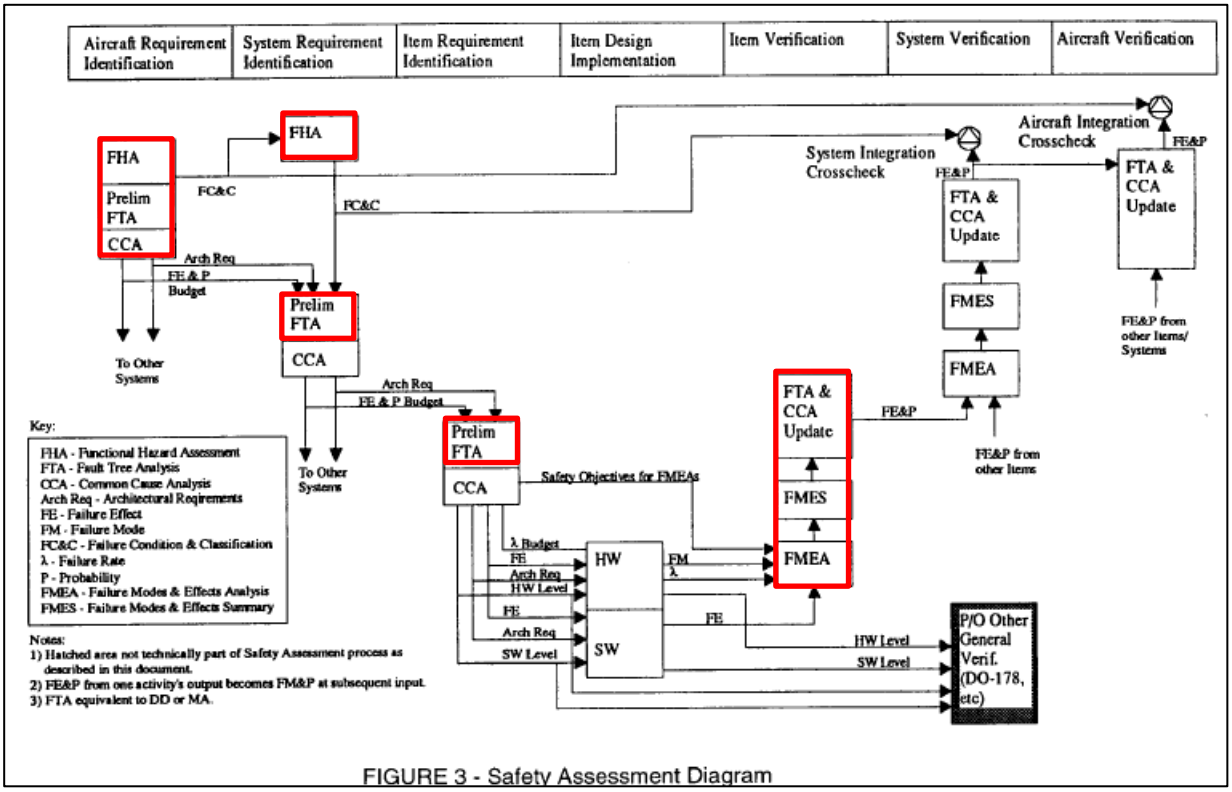


FIGURE 3 - Safety Assessment Diagram

Figure 102 Extrait des méthodes d'évaluation de l'ARP 4754

3.1. ANALYSE DE RISQUES FONCTIONNELLES AU NIVEAU DE L'AIRCRAFT FHA

Habituellement, on associe à l'analyse de risques, l'ensemble du profil de vie du système. Ici, par simplification, on ne considère que l'état de vol de l'hélicoptère.

La Table 21 est un extrait des fonctions opérationnelles considérées par l'ARP 4754 comme déclinaison de l'hélicoptère. La suite de l'étude se focalisera sur la fonction : « Control/Manage Fire protection ».

Table 21 Fonctions opérationnelles de l'hélicoptère pour la sûreté de fonctionnement

Helicopter Functions		
↳	Control Thrust	
↳	Control Flight Path	
↳	Determine Orientation	
↳	Determine Position and Heading	
↳	Control Helicopter on the Ground	
	↳	Determine Air/Ground transition
	↳	Decelerate Helicopter on the Ground
	↳	Control Helicopter Direction on the Ground
↳	Control Cabin Environment	
↳	Control Helicopter Engine	
	↳	Control power Engine
	↳	Control Engine Environment
	↳	**Control/Manage Fire protection**
	↳	Manage resources
↳	…	

L'« Aircraft FHA » identifie, à partir des fonctions opérationnelles et des états du cycle de vie, les conditions possibles des pannes et leurs caractérisations selon les effets, leur classification et parfois leur objectif de fiabilité. La Figure 103 est un extrait de l'« Aircraft FHA » qui concerne la fonction « garantir l'intégrité du système face au incendie ».

Fire detection System FHA (Functional Hazard Assessment)						
Function Helicopter Lvl	**Function System Lvl**	**Failure Condition ID**	**Failure condition (Hazard Description)**	**Effect of failure Condition on Helicopter/Crew**	**Classification**	**Obj/FH**
2 Ensure take off, flight and landing capabilities	**2.7** Ensure Fire protection	**FC1**	Fire in MGB compartment (Main Gear Box)	Would affect significantly the take-off, flight, and landing capabilities. Pilot would receive warning.	**Hazardous/Severe Major**	**N/A**
		FC2	Fire in Engine compartment	Could affect significantly the take-off, flight, and landing capabilities. Pilot would receive warning.	**Major**	**N/A**

Figure 103 Extrait de l'Aircraft FHA de l'hélicoptère

En complément, l'ARP recommande également pour la « system FHA » d'associer les conditions des pannes entre elles ou avec des évènements externes pour identifier les combinaisons de criticité « Major » ou « Catastrophic » (DAL A ou B). Le résultat d'analyse des conditions de panne est présenté à la Figure 104.

Fire detection System FHA (Functional Hazard Assessment)				
Combined FC	Effect of failure Condition on Helicopter/Crew	Classification	Obj/FH	Resultat
Loss of MGB fire warning signal generation	Loss of information on take-off, flight and landing capabilities	Major	1,00E-04 h^{-1}	Catastrophic
Loss of MGB fire warning display	Loss of information on take-off, flight and landing capabilities	Major	1,00E-04 h^{-1}	Catastrophic
Loss of engine fire warning signal protection	Loss of information on take-off, flight and landing capabilities	Major	1,00E-04 h^{-1}	Catastrophic
Loss of engine warning display	Loss of information on take-off, flight and landing capabilities	Major	1,00E-04 h^{-1}	Catastrophic
Complete loss of electrical power supplies to the Engine Control Box	Loss of information on take-off, flight and landing capabilities.	Major	1,00E-04 h^{-1}	Catastrophic

Figure 104 «Combined Failure condition» de l'Aircraft FHA

3.2. ANALYSE PRELIMINAIRE DES RISQUES - PASA

Des analyses PASA (Preliminary Aircraft Safety Analysis) ont été conduites. Cela a permis de construire des arbres de défaillance (cf. Figure 105 et Figure 106) comme une déclinaison des conditions des pannes du niveau « aéronef » (correspondant au niveau avion dans l'ARP 4754) au niveau système.

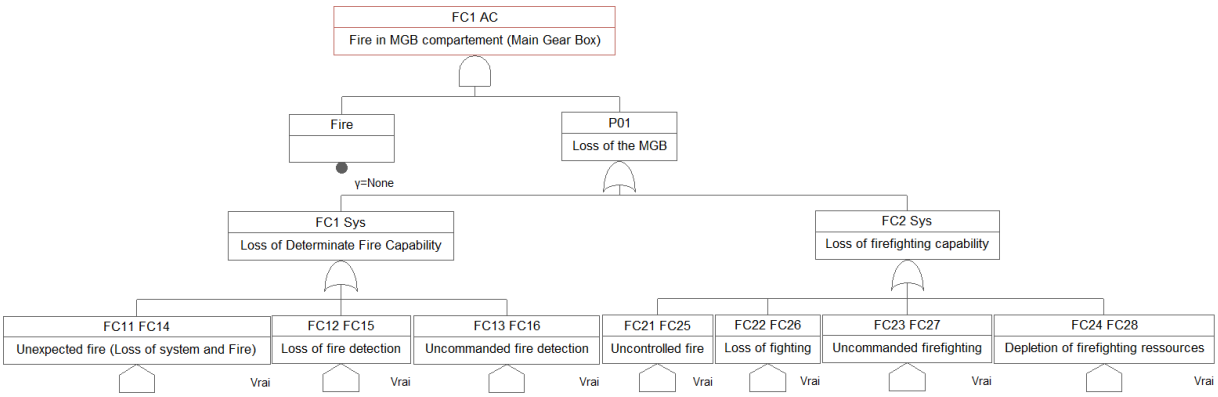


Figure 105 Arbre de défaillance de l'évènement "Feu dans le compartiment du rotor principal"

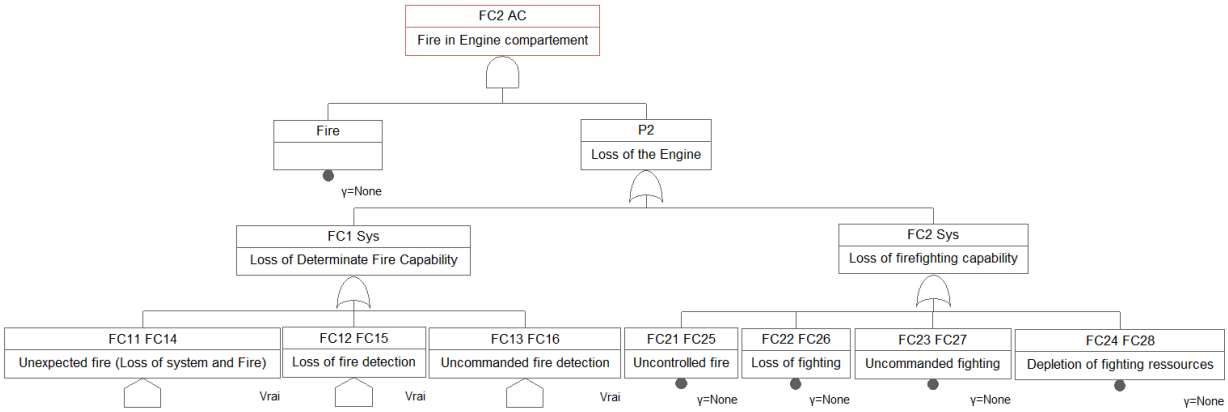


Figure 106 Arbre de défaillance de l'évènement "Feu dans le compartiment du moteur"

3.3. ANALYSE DE RISQUES FONCTIONNELLES AU NIVEAU DU SYSTEME - SYSTEM FHA

Une étude « System FHA » a été menée pour identifier les effets et la classification des conditions des pannes au niveau système. Les conditions des pannes sont issues de la PASA et concernent les fonctions opérationnelles présentées dans le diagramme Figure 107. Elles ont également été spécifié sous la forme d'un modèle AltaRica 3.0 [70], [71], cf. Figure 108.

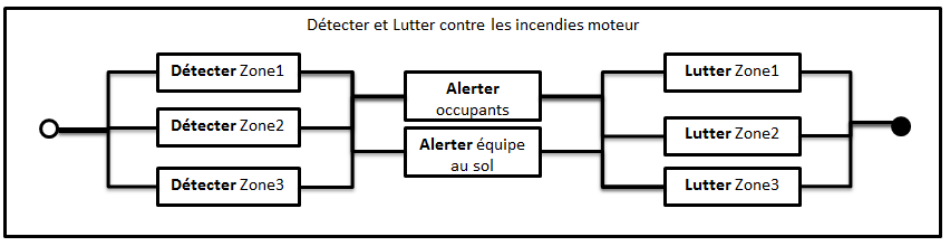


Figure 107 Diagramme Fonctionnel du système de détection incendie

```
Model1.alt
class eFunction
    Boolean working (init = true);
    Boolean input (reset = false);
    Boolean output (reset = false);
    event failure;
    transition
    failure : working -> working := false;
    repair : not working -> working := true;
end
```

```
Model2.alt
Import Model1.alt
block Artefact
    eFunction DetecterZ1, DetecterZ2, DetecterZ3;
    eFunction Alerter1, Alerter2;
    eFunction LutterZ1, LutterZ2, LutterZ3;
    Boolean input (reset = false);
    Boolean output (reset = false);
    assertion
    input := DetecterZ1.input;
    input := DetecterZ2.input;
    input := DetecterZ3.input;
    DetecterZ1.output := Alerter1.input;
    DetecterZ2.output := Alerter1.input;
    DetecterZ3.output := Alerter1.input;
    DetecterZ1.output := Alerter2.input;
    DetecterZ2.output := Alerter2.input;
    DetecterZ3.output := Alerter2.input;
    Alerter1.output := LuterZ1.input;
    Alerter1.output := LuterZ2.input;
    Alerter1.output := LuterZ3.input;
    Alerter2.output := LuterZ1.input;
    Alerter2.output := LuterZ2.input;
    Alerter2.output := LuterZ3.input;
end
```

Figure 108 Modèle AltaRica Système Fonctionnel du système de détection incendie

La FHA du système résultant de l'analyse de l'ingénieur est présentée Table 22.

Table 22 System FHA du système de détection incendie

Fire detection System FHA (Functional Hazard Assessment)						
Faillure Condition	**Faillure condition (Hazard Description)**	**Phase**	**Effect of failure Condition on Helicopter/Crew**	**Classification**	**Reference to Supporting Material**	**Verification**
FC1	Loss of Determinate Fire Capability	All	See Below	**Major**		
FC11	a. Unexpected fire (Loss of system and Fire)	TakeOff/ RTO/Landing/Fly	Crew is unable to Fly more. Crew shall eject himself as quick as possible.	**Catastrophic**	Emergency Landing procedures in case of fire	Helicopter FTA
FC12	b. Loss of fire detection	TakeOff/ RTO/Landing/Fly	Crew is unable to know is a fire appear. Crew shall try to land as soon as possible. Equipement need to be changed or redesign.	**Major**	Priority Landing procedures in case of fire	Helicopter FTA
FC13	c. Uncommanded fire detection	TakeOff/ RTO/Landing/Fly	Crew do not know there is'nt a fire. Crew shall try to land as soon as possible. Equipement need to be changed.	**Hazardous**	Priority Landing procedures in case of fire	Helicopter FMEA/FMES
FC14	d. Unexpected fire (Loss of system and Fire)	Taxi	Crew is unable to TakeOff. Crew shall evacuate in emergency as quick as possible.	**Major**	Priority Landing procedures in case of fire	Helicopter FTA
FC15	e. Loss of fire detection	Taxi	Crew is unable to TakeOff. Equipement need to be changed or redesign.	**Hazardous**	User documentation Logistics - Maintenance Phase FDS	Helicopter FMEA/FMES
FC16	f. Uncommanded fire detection	Taxi	Crew do not know there is'nt a fire. Crew shall stop engine as soon as possible. Equipement need to be changed.	**Minor**	User documentation Logistics - Maintenance Phase FDS	Helicopter FMEA/FMES
FC2	Loss of firefighting capability	All	See Below	**Catastrophic**		
FC21	a. Uncontrolled fire	TakeOff/ RTO/Landing/Fly	Crew is unable to Fly. Crew shall eject himself as quick as possible.	**Catastrophic**	Emergency Landing procedures in case of loss of fire detection	Helicopter FTA
FC22	b. Loss of fighting	TakeOff/ RTO/Landing/Fly	Crew shall try to land as soon as possible. Equipement need to be refuel.	**Major**	Priority Landing procedures in case of fire	Helicopter FTA
FC23	c. Uncommanded firefighting	TakeOff/ RTO/Landing/Fly	Crew is unable to TakeOff. Crew shall try to land as soon as possible. Equipement need to be changed or be refuel.	**Minor**	User documentation Logistics - Maintenance Phase FDS	Helicopter FMEA/FMES
FC24	d. Depletion of firefighting ressources	TakeOff/ RTO/Landing/Fly	Crew shall try to land soon. Equipement need to be refuel.	**Hazardous**	User documentation Logistics - Maintenance Phase FDS	Helicopter FMEA/FMES
FC25	a. Uncontrolled fire	Taxi	Crew is unable to TakeOff. Crew shall evacuate quickly. Equipement need to be refuel.	**Major**	User documentation Logistics - Maintenance Phase FDS	Helicopter FTA
FC26	b. Loss of fighting	Taxi	Crew is unable to TakeOff. Equipement need to be changed.	**Minor**	User documentation Logistics - Maintenance Phase FDS	Helicopter FMEA/FMES
FC27	c. Uncommanded firefighting	Taxi	Crew is unable to TakeOff. Equipement need to be changed and refuel.	**No Safety Effect**	User documentation Logistics - Maintenance Phase FDS	
FC28	d. Depletion of firefighting ressources	Taxi	Crew is unable to TakeOff, Equipement need to be refuel.	**No Safety Effect**	User documentation Logistics - Maintenance Phase FDS	

Les états du cycle de vie de l'hélicoptère et la classification sont résumés à la Figure 109.

Phase (lvl1)	Phase (lvl2)	Classification	
Takeoff	Optimal TakeOff	Catastrophic	DAL A
Takeoff	Climb with One Engine Inoperative (OEI)	Major	DAL B
RTO		Hazardous	DAL C
Fly	hovering flight	Minor	DAL D
Fly	vertical flight	No Safety Effect	DAL E
Fly	horizontal flight		
Fly	oblique flight		
Landing	Optimal Landing		
Landing	Climb with One Engine Inoperative (OEI)		
Taxi			

Figure 109 Niveau de classification et profil de vie du système

Les fonctions ont été allouées aux composants physiques et représentées Figure 110.

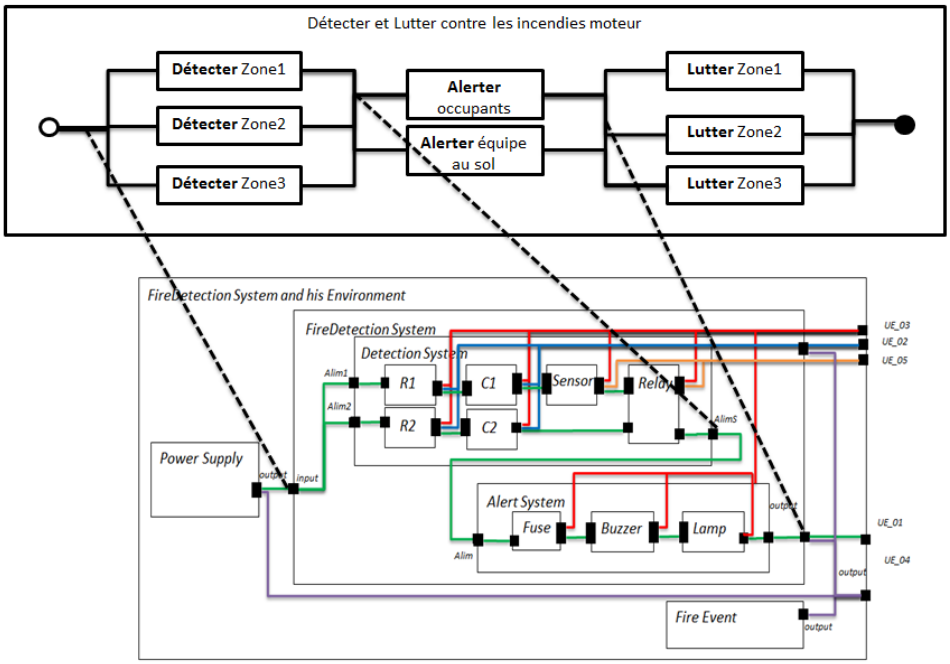


Figure 110 Allocation des fonctions sur les composants physiques

3.4. ANALYSE PRELIMINAIRE DES RISQUES AU NIVEAU DU SYSTEME- PSSA

Des analyses PSSA (Preliminary System Safety Analysis) ont été conduites. Elles ont permis de déduire à partir d'arbres de défaillance (ou de diagramme block de fiabilité) une déclinaison des conditions de panne de niveau système à des défaillances au niveau composants.

Les résultats obtenus sur deux conditions de panne relatifs à « Loss of Determinate Fire Capability » FC1 niveau Aircraft, sont présentés Figure 111 et Figure 112.

- PSSA du FC11 : « Unexpected fire (Loss of system and Fire) »

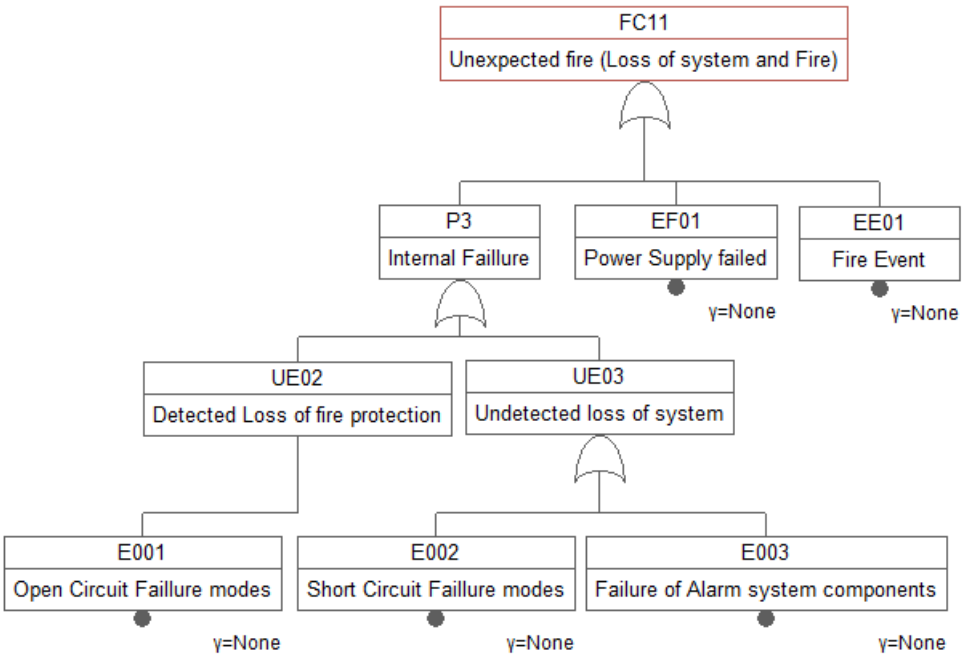


Figure 111 Arbre de défaillance "Feu non contrôlé"

- PSSA du FC12 : « Loss of fire detection »

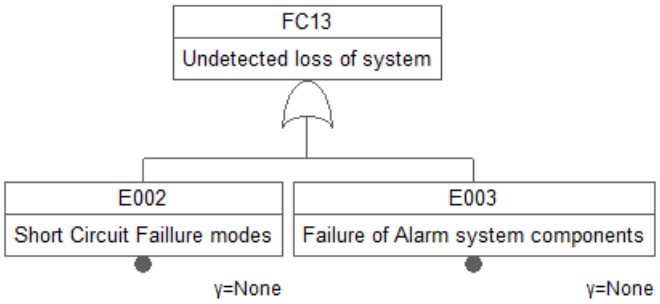


Figure 112 Arbre de défaillance "Défaillance du système non détectée"

3.5. ETUDE PROBABILISTE DE SURETE - ETUDE PSA

3.5.1. CONTEXTE

L'évaluation probabiliste de sureté (PSA) consiste à évaluer une architecture selon une liste d'exigences de sûreté de fonctionnement. La méthodologie pour mener une étude PSA doit respecter le Safety Program Plan [21].

Dans notre contexte, l'ingénieur de sûreté de fonctionnement doit évaluer le système de détection d'incendie (excluant la partie lutte incendie) pour attester de la conformité de la conception aux exigences du programme de l'hélicoptère. Le système concerné est non réparable. Seules les propriétés de fiabilité et de sécurité (safety) seront considérées.

Pour commencer une PSA, un ensemble de données d'entrée est requis. L'ingénieur de sûreté de fonctionnement recueille :

- L'architecture du système physique construite par l'architecte système. Hypothèse : l'architecte a utilisé le langage de modélisation SysML.

- L'hélicoptère FHA et les systèmes FHA (Functional Hazard Analysis) réalisés par l'ingénieur de sûreté de fonctionnement pendant les premières étapes du programme de sécurité et la liste des exigences qui en résulte.
- La description de l'environnement du système réalisée par l'architecte système ou l'ingénieur de sûreté de fonctionnement (en fonction de l'entreprise, le rôle peut être alloué à l'un ou l'autre).

Lorsque l'évaluation PSA commence, des analyses de sécurité antérieures ont déjà été effectuées sur l'hélicoptère. Elles fournissent de nombreux résultats, y compris les exigences de sûreté de fonctionnement qui doivent être validées par des analyses. Les exigences retenues sont les suivantes :

- Exigences issues de la FHA niveau hélicoptère (en référence au niveau « avion » dans l'ARP 4754).
    - Le système de détection d'incendie est DAL B,
        - Le système doit avoir un MTBF égal ou supérieur à 10 000 h^{-1}.
- Exigences de sécurité fonctionnelle provenant du système FHA.
    - FC1: feu incontrôlé inférieur ou égal à 1.10^{-9} h^{-1}
        - Exigence de sécurité: un incendie non contrôlé doit se produire moins de 10^{-5} h^{-1}
    - FC2: perte de protection contre l'incendie ≤ 1.10^{-5} h^{-1}
        - Exigence de sécurité : la perte de protection contre le feu détecté doit être inférieure à inférieur ou égal à 10^{-6} h^{-1}

          Remarque: il n'y a aucune exigence pour "perte de protection contre les incendies non détectés", il est considéré comme un événement inattendu.
    - FC3: protection anti-incendie non commandée inférieur ou égal à 1.10^{-4} h^{-1}
        - Exigence de sécurité: l'événement d'alarme non commandée doit se produire moins de inférieur ou égal à 10^{-4} h^{-1}
- Description de l'environnement du système
    - Pannes externes: EF_01 Perte de la source l'alimentation électrique inférieur ou égal à 1.10^{-5} h^{-1}
    - Evénements externes: EE_01 Evénement feu inférieur ou égal à 1.10^{-5} / h

Les exigences spécifiées par le programme de l'hélicoptère sont résumées dans la Table 23.

Table 23 Liste des exigences prises en compte lors de la PSA

Requirement's sources	**Requirement's type**	**Title** (link to Fire detection system Req)	**Objective values** (Failure/Hour)
Helicopter FHA, DAL B	Reliability	REQ_REL01 : Loss of the system - Lambda	10^{-4} h^{-1}
System FHA, FC11	Safety	REQ_SAF01 : Uncontrolled fire	10^{-9} h^{-1}
System FHA, FC12	Safety	REQ_SAF02 : Detected Loss of fire protection	10^{-6} h^{-1}
System FHA, FC13	Safety	REQ_SAF03 : Uncommanded alarm	10^{-4} h^{-1}

Les exigences sont attribuées à chaque zone surveillée du système. Par conséquent, il y a 12 exigences. Pour simplifier le cas d'étude, seul le moteur principal sera considéré (4 exigences).

3.5.2. APPLICATION DE L'EVALUATION

L'évaluation menée se déroule de façon itérative. Les différentes étapes sont les suivantes :

- Etape 1 : Traduction du modèle d'architecture du système ;
- Etape 2 : Définition des évènements redoutés ;
- Etape 3 : AMDEC et estimation de fiabilité ;
- Etape 4 : Construction des évènements redoutés et des comportements de défaillance ;
- Etape 5 : Modélisation et analyses ;
- Etape 6 : Interprétation des résultats.

Etape 1 : Traduction du modèle

La première étape du processus porte sur la reconstruction de l'architecture du système dans un formalisme dédié aux études de sûreté de fonctionnement. Cette étape est l'une des plus critiques car elle nécessite un apprentissage rigoureux du système de ses composants et de la documentation rattachée. Les ingénieurs de sûreté de fonctionnement risquent de faire des erreurs ou de mauvaises interprétations au cours de ce processus. Généralement, les informations collectées sont recueillies dans des feuilles de calcul. Le modèle résultant manque de visibilité et il n'est pas entièrement aligné sur le modèle initial de la conception du système.

Dans notre cas, le modèle d'architecture système est directement traduit par la transformation de modèles (cf. Chapitre VI). Cette technique est une première étape pour assurer un premier niveau de cohérence entre des modèles répondant à différentes préoccupations. Pour cela, le framework Sophia [74], [73] sera utilisé.

Après cette activité, l'ingénieur de sûreté de fonctionnement dispose de la structure de l'architecture, de la liste des composants et des interactions entre composants (leurs ports définissant les entrées et les sorties). À ce stade, les composantes du modèle sont représentées comme des boîtes noires.

Etape 2 : Définition des évènements redoutés

L'ingénieur de sûreté de fonctionnement doit définir quels événements doivent être évalués : ces événements s'appellent « événements redoutés ». Cette étape est cruciale car elle définit les propriétés qui seront qualifiées par l'analyse PSA. Le résultat de cette étape est une liste d'événements redoutés attachés aux exigences à évaluer.

Les « événements redoutés » sont une déclinaison des exigences destinées à exprimer les critères d'acceptabilité du système. La Table 24 et la Figure 113 montrent la liste des événements redoutés identifiés et leurs relations.

Table 24 Liste des évènements redoutés

UE_Number	**Unexpected Events (UE)**
UE_01	Loss of the system
UE_02	Detected Loss of fire protection
UE_03	Undetected loss of system
UE_04	Uncontrolled fire
UE_05	Uncommanded alarm

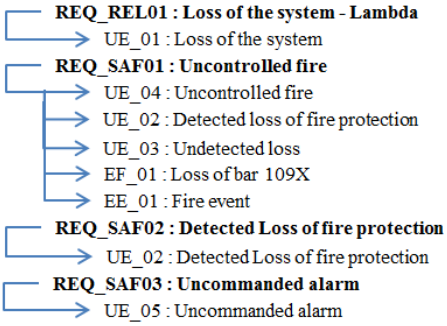


Figure 113 Relations entre les exigences et les évènements redoutés

Etape 3 : Estimation de la fiabilité et AMDEC

L'étape consiste à estimer la fiabilité des éléments du système. Il faut d'abord identifier les modes de défaillance des composants et leurs répartitions dans une AMDEC enrichie de quelques propriétés.

L'estimation de la fiabilité des composants électroniques sera définie à partir du guide MIL HDBK 217 [45]. Ce manuel est très utilisé, bien que ces estimations deviennent de plus en plus obsolètes, [136]. Toutefois ces hypothèses sont largement acceptables pour nos besoins. De même, une loi exponentielle est utilisée pour caractériser la fiabilité intrinsèque de chaque composant électronique du système.

Le cadre suivant détaille, sur un composant, les calculs permettant d'obtenir l'estimation du taux de défaillance λ.

Capacitor C1: Multilayer ceramic capacitor 100nF, operating voltage: 3,3V, maximum 50V, series resistance factor: 1. Capacitors are manufactured with a standard quality (P level).
This is CDR Ceramic, Chip, Est.Rel in MIL HDBK 217F – "10.11 Capacitors, Fixed, Ceramic, temperature Compensating and Chip".

$$\lambda_p = \lambda_b . \pi_{CV} . \pi_Q . \pi_E \text{ Failures}/10^6 \text{ Hours}$$

$\lambda_b = 0.0145$; $\pi_{cv} = 2.35$; $\pi_q = 0.30$; $\pi_e = 8$ (A_{IF}); Therefore $\boldsymbol{\lambda_p = 8.18 . 10^{-8}\ h^{-1}}$

Suivant le même raisonnement, les estimations des fiabilités des composants sont calculées. Ils sont résumés dans la Table 25.

Table 25 Fiabilités intrinsèques calculées des composants

Component	**λ_p (en h^{-1})**	**Component**	**λ_p (en h^{-1})**
Capacitor C1	$8{,}18 .10^{-8}$	Lamp	$1{,}50 .10^{-6}$
Capacitor C2	$9{,}26 .10^{-8}$	Detector	$3{,}60 .10^{-5}$
Resistor R1	$1{,}20 .10^{-7}$	Buzzer	$8{,}00 .10^{-7}$
Resistor R2	$8{,}14 .10^{-8}$	Relay	$2{,}10 .10^{-6}$
Fuse	$1{,}60 .10^{-7}$	MTBF = $1/\Sigma\lambda_T$ = 24 400 hours	

La fiabilité calculée est intrinsèque à chaque composant. Pour évaluer la répartition de la fiabilité sur les modes de défaillance interne des composants, l'UTE-C80 810 [49] fournit une estimation de la distribution des modes de défaillance d'un composant. La Table 26 résume l'information extraite de ce référentiel.

Failure Mode Distribution		
Type of Item	**Failure mode**	**Failure mode ratio**
Resistor, CTN	Open Circuit	40 %
	Drift	60 %
Caramic Capacitor	Open Circuit	30 %
	Short Circuit	70 %
Diode	Open Circuit	20 %
	Short Circuit	80 %
Transistor	Open Circuit	15 %
	Short Circuit	85 %
Detector	Loss	33 %
	Uncommanded Function	67 %
Other parts (connector, fuse, switch)	Loss	100 %

Table 26 Distribution de la défaillance sur les modes de défaillance

Une AMDEC est une méthode permettant de détailler et de qualifier les modes de défaillance des composants et leur criticité. Il existe plusieurs types d'AMDEC (produit, processus, système,…). L'AMDEC système, présentée Table 27, est axée sur les composants et leurs comportements dysfonctionnels. Elle fournit, pour chaque composant, des modes de défaillance, leur(s) effet(s) (effet local ou effet système) et des données complémentaires. Le taux de défaillance des composants est distribué sur les modes de défaillance respectifs.

Table 27 Récapitulatif des défaillances, extrait de l'AMDEC (FMES)

Component	**Failure rate component**	**FM ref number**	**Failure mode**	**Distribution**	**System Effect**	**Failure rate**
Capacitor C1	$8{,}18.10^{-08}$	1.1	Open Circuit	30%	Detected loss	$2{,}46.10^{-08}$
		1.2	Short Circuit	70%	Undetected loss	$5{,}73.10^{-08}$
Capacitor C2	$9{,}26.10^{-08}$	2.1	Open Circuit	30%	Detected loss	$2{,}78.10^{-08}$
		2.2	Short Circuit	70%	Undetected loss	$6{,}48.10^{-08}$
Resistor R1	$1{,}20.10^{-07}$	3.1	Open Circuit	40%	Detected loss	$4{,}78.10^{-08}$
		3.2	Drift	60%	Undetected loss	$7{,}18.10^{-08}$
Resistor R2	$8{,}14.10^{-08}$	4.1	Open Circuit	40%	Detected loss	$3{,}25.10^{-08}$
		4.2	Drift	60%	Undetected loss	$4{,}88.10^{-08}$
Fuse	$1{,}60.10^{-07}$	5.1	Loss	100%	Undetected loss1	$1{,}60.10^{-07}$
Lamp	$1{,}50.10^{-06}$	6.1	Loss	100%	Undetected loss1	$1{,}50.10^{-06}$
Sensor	$3{,}60.10^{-05}$	7.1	No output	33%	Undetected loss1	$1{,}19.10^{-05}$
		7.2	Uncommanded Function	67%	Uncommanded alarm1	$2{,}41.10^{-05}$
Buzzer	$8{,}00.10^{-07}$	8.1	Loss	100%	Undetected loss1	$8{,}00.10^{-07}$
Relay	$2{,}10.10^{-06}$	9.1	Open Circuit	95%	Undetected loss1	$2{,}00.10^{-06}$
		9.2	Short Circuit	5%	Uncommanded alarm1	$1{,}00.10^{-07}$

Etape 4 : Construction des évènements redoutés et du comportement de défaillance

Avant de quantifier les événements redoutés, il faut identifier les modes de défaillance qui les génèrent. Les «événements inattendus» sont exprimés par logique booléenne (« + » exprimant un « OU » et « . » exprimant un « ET ») à partir de «modes de défaillance interne» (FMj.k), d'«événements externes» (EE) et de «défaillances externes» (EF).

UE_01 = FM1.1 + FM1.2 + FM2.1 + FM2.2 + FM3.1 + FM3.2 + FM4.1 + FM4.2 + FM5.1 + FM6.1 + FM7.1 + FM7.2 + FM8.1 + FM9.1 + FM9.2
UE_02 = FM1.1 + FM2.1 + FM3.1 + FM4.1
UE_03 = FM1.2 + FM2.2 + FM3.2 + FM4.2 + FM5.1 + FM6.1 + FM7.1 + FM8.1 + FM9.1
UE_04 = EE_01 . (EF_01 + UE_02 + UE_03)
= EE_01 . (EF_01 + FM1.1 + FM1.2 + FM2.1 + FM2.2 + FM3.1 + FM3.2 + FM4.1 + FM4.2 + FM5.1 + FM6.1 + FM7.1 + FM8.1 + FM9.1)
UE_05 = FM7.2 + FM9.2

Etape 5 : modélisation et analyse

Pour effectuer l'évaluation sur chaque évènement redouté, les modèles seront conçus avec un langage de modélisation dédié à l'analyse de sûreté de fonctionnement : AltaRica 3.0 [70], [71].

Pour évaluer les exigences, les analyses menées en amont ont été rédigées dans un modèle AltaRica 3.0. C'est le compilateur d'arbres de défaillance qui est utilisé pour construire les arbres correspondant à chaque événement redouté dans le format d'échange de modèle Open-PSA. Après cela, xFTA [137] sera utilisé pour calculer la probabilité de l'événement sommet.

Le formalisme AltaRica 3.0 permet la description des composants et de l'architecture système d'une manière simple.

Le bloc élémentaire, Figure 114, présente la description de Résistance R1 dans le formalisme AltaRica 3.0. Il peut décrire les composants, les entrées, les sorties, les modes de défaillance et les comportements de défaillance. Dans l'exemple ci-dessous, Resistor1 comporte deux modes de défaillance FM31 et FM32 et les comportements dysfonctionnels associés lors de l'occurrence d'une défaillance, enfin la logique de propagation de la défaillance (partie « assertion »). Une fois les composants modélisés, les flux de propagation des défaillances sont définis entre ces composants. Le formalisme permet ainsi la modélisation du système et des sous-systèmes hiérarchisés.

```
//------------the Resistor---------
class Resistor1
//state declaration
   Boolean FM31 (init = false);
   Boolean FM32 (init = false);
//flow declaration
   event Resistor_ShortCut_occurs ( delay = exponential( 0.0000000478 ) ) ;
   event Resistor_OpenCircuit_occurs ( delay = exponential( 0.0000000718 ) );
   transition
      Resistor_ShortCut_occurs: Resistor_FailureMod1==false->Resistor_FailureMod1:=true;
      Resistor_OpenCircuit_occurs: Resistor_FailureMod2==false->Resistor_FailureMod2:=true;
   assertion
      s1 := (not(FM31) and not(FM31) and e1);
end
```

Figure 114 Description de la résistance R1 en AltaRica3.0

La Figure 115 est une vue graphique (S2ML, [71]) représentant l'architecture du système telle qu'elle est implémentée en AltaRica 3.0. Les connexions entre les blocs reflètent les flux de propagation des défaillances entre les composants du système.

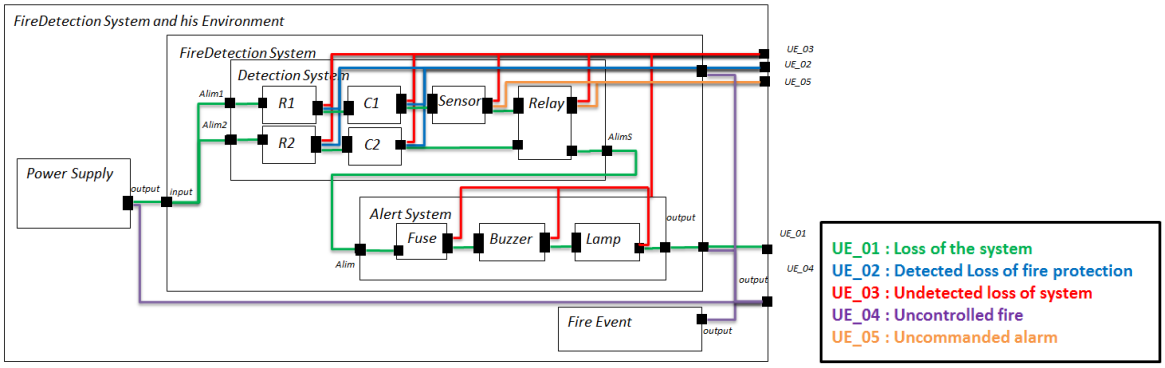


Figure 115 Modélisation en S2ML du système

En utilisant le compilateur d'arbres de défaillance des outils AltaRica 3.0, cinq arbres de défaillance ont été obtenus, un par évènement redouté. Ils sont stockés dans le formalisme

Open-PSA [135]. La Figure 116 représente l'arbre de défaillance de l'évènement redouté 1 « Perte du système ». Pour des raisons de lisibilité, l'arbre généré est présenté sous sa forme développée. Le même exercice est effectué sur les quatre autres arbres.

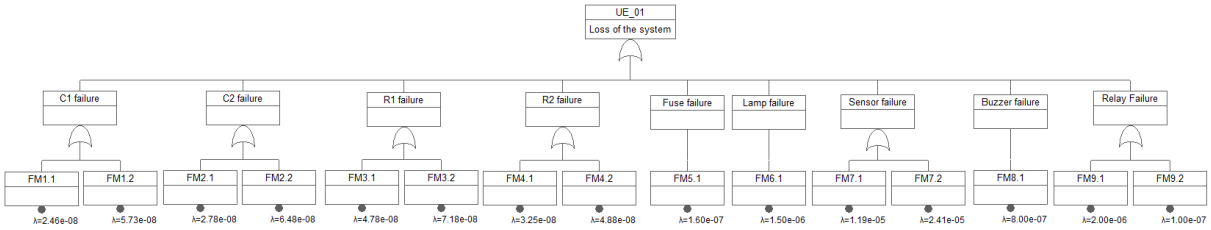


Figure 116 Arbre de défaillance résultant du modèle AltaRica 3.0

Pour chaque arbre de défaillance, l'outil xFTA [137] est utilisé pour calculer les probabilités de défaillances des évènements redoutés. XFTA permet de mener des évaluations probabilistes comme :

- Le calcul de la probabilité de l'événement sommet sur différents temps de mission ;
- Le calcul des facteurs d'importance pour les événements de base (Birnbaum / Facteur d'importance marginale, Facteur d'importance critique, Facteur d'importance du diagnostic, Valeur de réduction des risques, Valeur du succès) ;
- L'analyse de sensibilité au moyen de la simulation de Monte-Carlo ;
- Le calcul des niveaux d'intégrité de sûreté pour un mode de défaillance à faible demande et à demande élevée ou continue pour les systèmes liés à la sécurité, conformément à la norme de sécurité CEI 61508 [72] et ses déclinaisons.

Etape 6 : Interprétation des résultats

La fiabilité des cinq événements redoutés est obtenue, en exécutant xFTA [137], selon différents temps de mission. Dans le cas d'étude, la durée de mission est de 1000 h. La Table 28 présente les événements calculés par xFTA et montre les résultats correspondants.

Table 28 Résultats d'occurrence des évènements redoutés

UE Number	**Unexpected Events (UE)**	Value
UE_01	Loss of the system	$4{,}09.10^{-5}\ h^{-1}$
UE_02	Detected Loss of fire protection	$1{,}33\ .10^{-7}\ h^{-1}$
UE_03	Undetected loss of system	$1{,}66\ .10^{-5}\ h^{-1}$
UE_04	Uncontrolled fire	$8{,}51\ .10^{-8}\ h^{-1}$
UE_05	Uncommanded alarm	$2{,}42\ .10^{-5}\ h^{-1}$

A partir des valeurs calculées, il est possible de comparer celles-ci aux exigences correspondantes. Les résultats obtenus sont synthétisés dans la Table 29.

Table 29 Interprétation des résultats avec les exigences

Type of requirement	**Title** (link to Fire detection system req)	**Values (Objective)**	**Values (Calculated)**	**Requirement Validity**
Reliability	REQ_REL01 : Loss of the system	$10^{-4}\ h^{-1}$	$4{,}09.10^{-5}\ h^{-1}$	**True**
Safety	REQ_SAF01 : Uncontrolled fire	$10^{-9}\ h^{-1}$	$8{,}51.10^{-8}\ h^{-1}$	**False**
Safety	REQ_SAF02 : Detected Loss of fire protection	$10^{-6}\ h^{-1}$	$1{,}33.10^{-7}\ h^{-1}$	**True**
Safety	REQ_SAF03 : Uncommanded alarm	$10^{-4}\ h^{-1}$	$2{,}42\ .10^{-5}\ h^{-1}$	**True**

Il est important de noter que l'objectif de sûreté REQ_SAF01 de 1. 10^{-9} n'est pas respecté. Par conséquent, le système ne peut être validé. Une modification doit être menée pour satisfaire les exigences et plusieurs alternatives peuvent être envisagées :

- Mettre en place un contrôle préventif ;
- Faire évoluer l'architecture du système ;
- Utiliser des composants plus fiables.

Les ingénieurs ont choisi d'adapter l'architecture du système et de le soumettre à proposition à l'architecte du système. Mais avant de le lui soumettre, il faut vérifier le respect de la nouvelle architecture aux exigences. L'architecte système est informé des résultats obtenus et prévenu de cette démarche.

Une redondance d'ordre 2 est proposée en ajoutant une barrière sur l'ensemble du système. Puis le système est réévalué.

3.5.3. APPLICATION D'UNE SECONDE EVALUATION

À la suite de la première analyse PSA, des recommandations ont été envoyées à l'architecte système. Ces recommandations ont été acceptées et prises en compte. L'architecture du système est modifiée en conséquence. Les modèles employés dans la première analyse sont adaptés à la nouvelle version de l'architecture du système. Notons que la deuxième itération est plus rapide à mener car les modèles et les données sont déjà structurés. Il faut prendre soin de ne pas oublier de dépendances entre les données lors de la mise à jour des modèles.

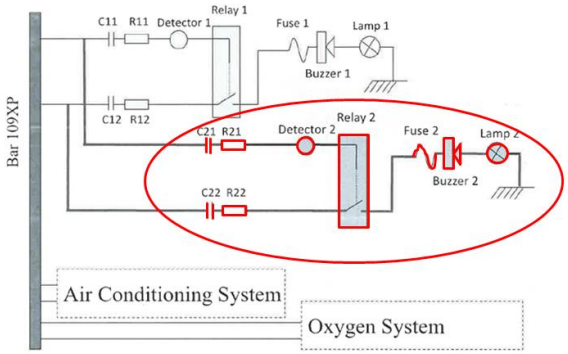


Figure 117 Schéma électrique du système avec redondance

Dans ce nouveau modèle, Figure 117, une redondance a été introduite. L'architecture du système a également été adaptée, en conséquence, dans le formalisme AltaRica 3.0 correspondant, cf. Figure 118.

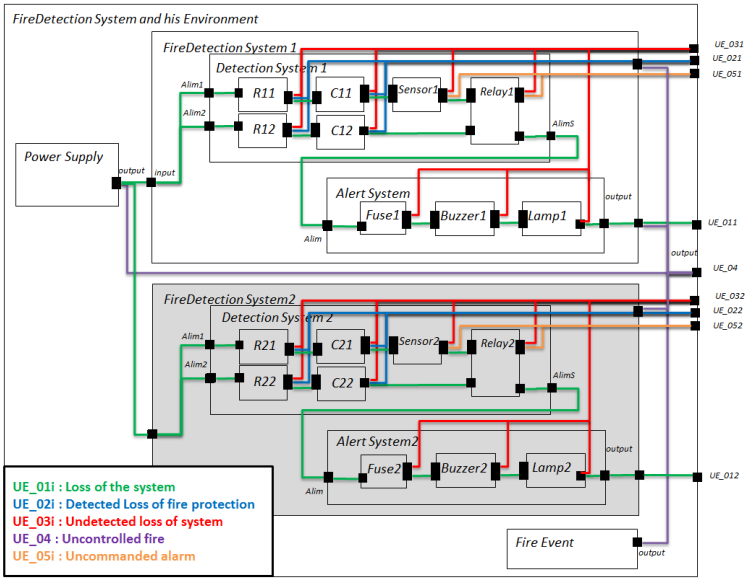


Figure 118 Modélisation du système en S2ML avec une redondance

Mise à jour des événements redoutés

L'expression des événements redoutés de l'analyse PSA précédente est conservée, mais l'architecture oblige à inclure un second niveau de décomposition des événements, cf. Table 30.

Table 30 Relation entre les évènements redoutés

UE_ Number	Unexpected Events (UE)	UE_ Number	Unexpected Events (UE)	Unexpected Event constructions
UE_01	Loss of the system	UE_011	Loss of the system1	UE_01 = UE_011 + UE_012
		UE_012	Loss of the system2	
UE_02	Detected loss fire protection	UE_021	Detected Loss 1 of fire protection	UE_02 = UE_021 + UE_022
		UE_022	Detected Loss 2 of fire protection	
UE_03	Undetected loss of system	UE_031	Undetected loss 1 of system	UE_03 = UE_031 + UE_032
		UE_032	Undetected loss 2 of system	
UE_04	Uncontrolled fire	UE_041	Uncontrolled fire 1	UE_04 = EE_01 . (EF_01 + UE_02 + UE_03)
		UE_042	Uncontrolled fire 2	
UE_05	Uncommanded alarm	UE_051	Uncommanded alarm 1	UE_05 = UE_041 + UE_042
		UE_052	Uncommanded alarm 2	

Estimation de la fiabilité

Les nouveaux composants sont toujours estimés à l'aide du guide MIL HDBK 217 [45]. Les composants étant simplement dupliqués, leurs caractéristiques intrinsèques sont les mêmes et l'estimation de leur fiabilité ne change pas.

Mise à jour d'AMDEC

L'AMDEC est complètement similaire à la précédente, mais il y a désormais des composants en double. Par conséquent, deux fois plus de modes de défaillance en résultent. Mais les composants, les taux de défaillance, la distribution dans le temps et la fiabilité des modes de défaillance ne sont pas affectés par la mise à jour.

En utilisant la même chaîne d'outils que pour la première analyse PSA, on obtient les résultats synthétisés dans la Table 31 et la Table 32.

Table 31 Résultats sur l'occurrence des évènements redoutés avec une redondance

UE_ Number	Unexpected Events (UE)	Value
UE_01	Loss of the system	$8{,}19.10^{-5}$ h^{-1}
UE_02	Detected loss fire protection	$2.66.10^{-7}$ h^{-1}
UE_03	Undetected loss of system	$5.32.10^{-5}$ h^{-1}
UE_04	Uncontrolled fire	$1{,}02.10^{-9}$ h^{-1}
UE_05	Uncommanded alarm	$4{,}82\ .10^{-5}$ h^{-1}

Table 32 Seconde interprétation des résultats avec les exigences

Requirement type	Title (link to Fire detection system req)	Values (Objective)	Values (Calculated)	Validity
Reliability	Loss of the system	10^{-4} h^{-1}	$8{,}19.10^{-5}$ h^{-1}	**True**
Safety	Uncontrolled fire	10^{-9} h^{-1}	$1{,}02.10^{-9}$ h^{-1}	**True**
Safety	Detected Loss of fire protection	10^{-6} h^{-1}	$2.66.10^{-7}$ h^{-1}	**True**
Safety	Uncommanded alarm	10^{-4} h^{-1}	$4{,}82\ .10^{-5}$ h^{-1}	**True**

Il est important de noter que toutes les exigences sont validées. Par conséquent, l'architecte système est informé des nouvelles propriétés de l'architecture du système. Si une modification devrait être faite sur l'architecture, l'analyse PSA devrait être menée à nouveau.

3.6. RESUME DES ANALYSES

L'architecte système a fait évoluer le modèle d'architecture système en incluant les redondances nécessaires. Ce cas d'étude a montré un extrait des études qui ont pu être menées. Il témoigne pas à pas de l'usage des concepts, des méthodes, des points de vue et des référentiels.

Ces travaux et les données produites seront réemployés pour illustrer des applications de synchronisation dans le Chapitre V.

Chapitre V APPLICATION DE LA METHODOLOGIE SUR LE CAS D'ETUDE

Ce chapitre présente l'application des activités de la méthodologie de synchronisation (cf. Chapitre III) sur le cas d'étude du Chapitre IV. Le but est d'illustrer, sur le même exemple, le déroulement complet des 5 activités de la méthodologie. Cinq points de synchronisation y sont étudiés à différentes étapes des processus et à différents niveaux d'abstraction.

Dans le contexte du cas d'étude, le directeur général (DG) de l'Equipementier A sollicite un responsable de synchronisation pour mettre en œuvre une démarche collaborative de synchronisation de modèles entre des disciplines d'ingénierie. Les étapes effectuées par le responsable synchronisation sont présentées, tout au long de ce chapitre.

1. MISE EN PRATIQUE DE LA DEFINITION DES PRINCIPES D'INTERACTIONS

1.1. METHODE « CHOIX DES DISCIPLINES D'INGENIERIE »

Pour mettre en œuvre une solution de synchronisation de modèles, le DG a choisi d'améliorer l'organisation des interactions entre l'équipe d'architecture système et l'équipe d'analyse de sûreté de fonctionnement. Il oriente cette action parce que les retours d'analyse et d'évaluation des ingénieurs de sûreté de fonctionnement, dans son entreprise, ne parviennent pas assez tôt dans le cycle de développement des systèmes. Il aimerait que les architectes puissent tenir compte des études de sûreté de fonctionnement dès les premières étapes du processus de conception. Il pense qu'investir des ressources et du temps en phase amont permettrait un gain conséquent en termes de coût, de délai et de qualité pour les étapes qui suivent.

Conception d'architecture :

- Système étudié : système de détection et lutte incendie embarqué dans un hélicoptère de combat
- Missions : l'architecte système est en charge de trouver une solution d'architecture répondant aux besoins spécifiés du système à concevoir. Pour cela, il peut utiliser le formalisme qu'il souhaite.
- Préoccupations : les objectifs du système, l'architecture du système, l'adéquation de l'architecture pour atteindre les objectifs, la faisabilité de la construction et du déploiement du système.

L'équipe doit respecter les standards qui lui sont dédiés : l'ISO 42010 [8], ISO 42020 [31] et l'ISO 42030 [32].

Sûreté de fonctionnement :

- Système étudié : système de détection et lutte incendie embarqué dans un hélicoptère de combat

- Missions : l'ingénieur en sûreté de fonctionnement doit apprécier l'aptitude d'un système à remplir une ou plusieurs fonctions requises dans des conditions données à un instant donné. Son activité englobe principalement les composantes de fiabilité, de maintenabilité, de disponibilité et de sécurité. Il doit mesurer la conformité de ces performances calculées ou estimées avec les exigences du système afin de garantir un niveau de risque acceptable.
- Préoccupations : les risques et impacts potentiels du système pour les parties prenantes tout au long de son cycle de vie ; le comportement dysfonctionnel du système et l'évaluation de la performance du système en termes de fiabilité, de maintenabilité, de disponibilité et de sécurité.

L'équipe doit respecter les standards et les objectifs de certification qui la concernent.

1.2. METHODE « DEFINITION OPERATIONNELLE DU PROJET DE SYNCHRONISATION »

Pour définir l'organisation du projet et les interactions avec les parties prenantes, un diagramme de cas d'utilisation en UML [26] a été réalisé Figure 119. Il représente les cas d'utilisation par des ovales, les parties prenantes par des acteurs et leurs interactions par des relations d'association.

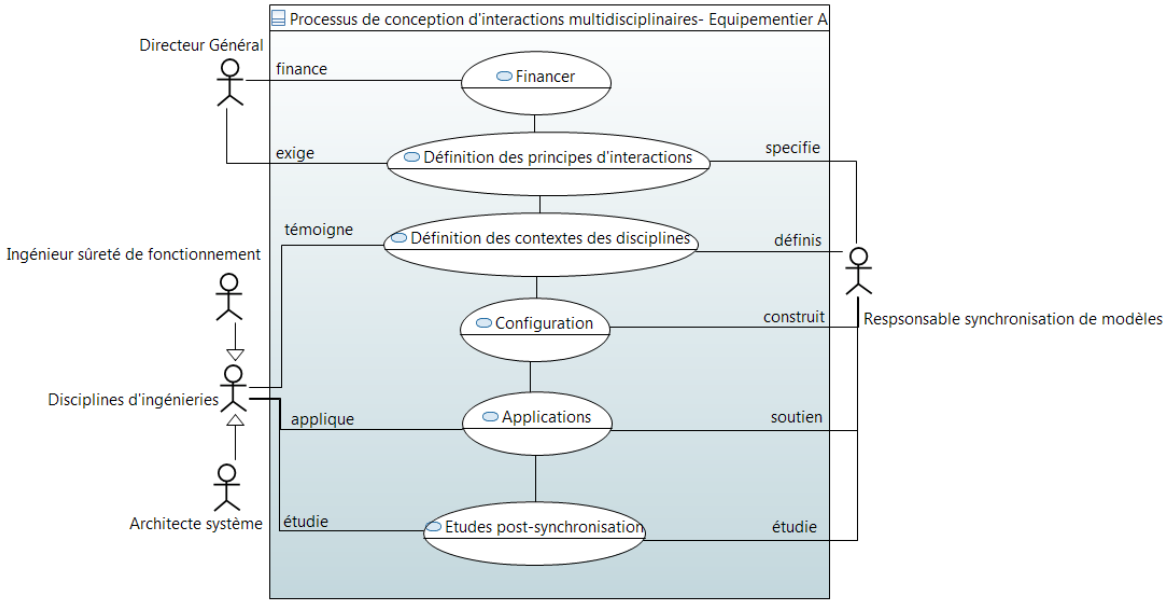


Figure 119 Diagramme UML de cas d'utilisation du processus de conception de synchronisation de modèles

1.3. METHODE « DEFINITION DE L'ETAT ACTUEL DES INTERACTIONS »

Des réunions avec les acteurs des deux disciplines d'ingénierie ont permis de clarifier le déroulement de leurs interactions. Il en ressort la synthèse suivante :

« A chaque projet, les deux équipes tentent d'interagir entre elles pour se mettre d'accord avant de débuter leurs études, cependant elles sont confrontées à des difficultés. Les termes employés par chacune des équipes sont assez proche, mais ils ne relèvent pas exactement des mêmes notions. Les équipes ne se comprennent pas toujours du fait des subtilités fondamentales qui se cachent dernières les éléments traités.

Dans un projet habituel, environ 5 réunions sont organisées (dont une en avant-projet). Pourtant les disciplines ne ressentent pas de réel bénéfice bien qu'elles partagent l'envie d'intensifier leurs interactions.

Les architectes souhaiteraient des retours bien plus en amont. Or, aujourd'hui ils sollicitent les ingénieurs de sûreté de fonctionnement tardivement dans le processus, souvent lors de la conception de l'architecture physique alors que des discussions pourraient être entamées sur l'architecture fonctionnelle.

Les ingénieurs en sûreté de fonctionnement communiquent avec les architectes systèmes pour comprendre puis évaluer la structure globale du système. Ils sont sollicités de plus en plus tôt dans les projets pour des analyses et des évaluations sur des solutions encore en évolution. Persuadée qu'il est nécessaire de travailler plus en amont, l'équipe semble se décourager progressivement sur ses capacités à traiter toutes les demandes d'études. »

1.4. METHODE « DEFINITION DE L'ETAT CIBLE »

Le responsable de synchronisation lit les principes généraux d'interactions et fait une sélection parmi des principes optionnels qui lui semblent pertinents dans le contexte du projet à traiter.

- Global :
  - Suivre une démarche collaborative ;
  - Guider et faciliter la communication ;
  - Centrer la synchronisation sur les architectures du système ;
  - Garantir un niveau de cohérence ;
  - Faciliter la gestion de la complexité ;
  - Permettre des interactions unidirectionnelles ou bidirectionnelles du contenu des modèles **(non retenu)**.
- Processus :
  - Renforcer les relations entre domaines d'ingénierie durant les étapes amont du projet **(retenu)** ;
  - Favoriser l'investissement en étape amont, pour faciliter les interactions entre domaines d'expertise **(retenu)**.
- Modèles :
  - Mettre en cohérence des modèles hétérogènes ;
  - Permettre l'interopérabilité des modèles ;
  - Origines des modèles connues ou inconnues, communes ou non **(non retenu)**.
- Décision :
  - Ouvrir/Renforcer/Imposer le dialogue entre les disciplines d'ingénierie choisies ;
  - Limiter les jugements/biais dans les études ;
  - Gérer/limiter/contrôler la prise de décision des disciplines d'ingénierie **(retenu)**.
- Résultat :
  - Apporter une justification de la mise en cohérence des études.

Il définit également des principes spécifiques :

- Processus :
  - PS1 - Les applications de la synchronisation de modèles devront être discutées en réunion avec les architectes systèmes et les ingénieurs sûreté de fonctionnement.

- Modèles :
  - PS2 - Les modèles utilisés devront être implémentés à partir de langages ou de métamodèles définis à l'avance dans une bibliothèque ;
  - PS3 – Les applications de synchronisation de modèles traiteront uniquement de la mise en cohérence des éléments et des relations de structuration qui caractérisent l'architecture du système.

Le responsable de synchronisation transmet aux disciplines d'ingénierie et au DG tous les principes retenus pour les informer des objectifs du projet.

1.5. METHODE « EVALUATION DE L'ADEQUATION DU BESOIN AVEC L'APPROCHE »

La Table 33 détaille pour chaque principe de synchronisation PS, les actions que le responsable de synchronisation mènera durant les étapes de la méthodologie.

Table 33 Evaluation de l'adéquation du besoin avec l'approche

Activité	Définition des principes d'interactions	Définition des contextes d'ingénierie	Configuration de synchronisation	Application de la synchronisation	Suivi et évolution de la cohérence
PS1				La synchronisation se réalisera en salle de réunion avec les deux équipes.	
PS2		Définir les bibliothèques des langages et métamodèles candidats.	Les mappings seront définis selon les concepts pivots d'architecture et les bibliothèques des langages et métamodèles.		
PS3		Vérifier que les langages et les modèles produits par les disciplines traduisent bien l'architecture du système.	Les métamodèles pivots ne contiendront que des propriétés faisant référence à l'architecture du système.		

Etude complémentaire : Aucune étude complémentaire n'a été effectuée.

2. MISE EN PRATIQUE DE LA DEFINITION DES CONTEXTES D'INGENIERIE

2.1. METHODE « CONTEXTES D'INGENIERIE »

La méthode « **Définition des contextes d'ingénierie »** a été adoptée. Celle-ci a permis de définir les processus, les activités, les méthodes et les points de vue utilisés par les équipes d'architecture système et de sûreté de fonctionnement de l'Equipementier A.

Les processus sont appliqués sur le cas d'étude du système de détection incendie. Pour chacune de ces activités, les méthodes et les points de vue associés ont été identifiés puis représentés dans la Figure 120.

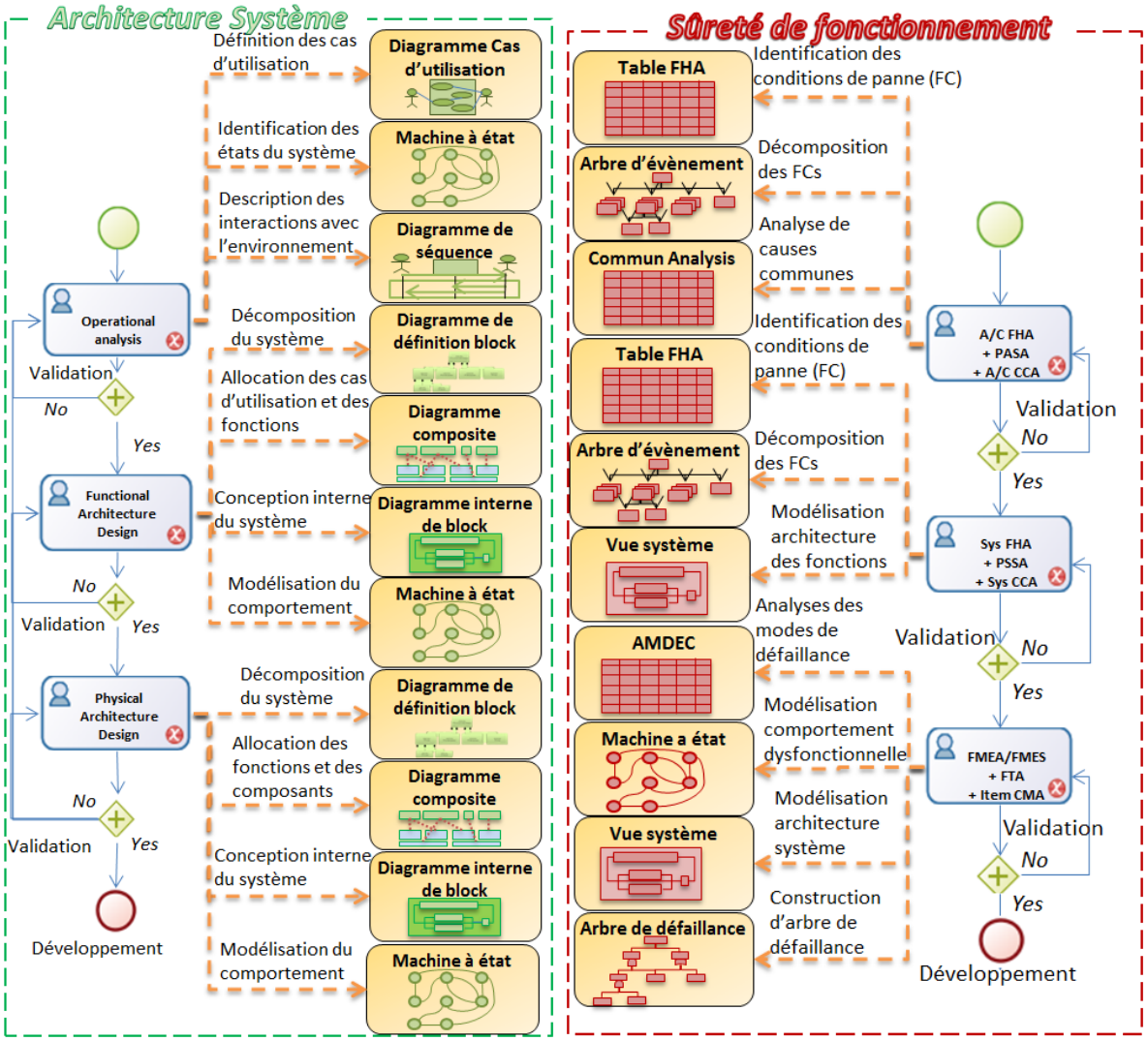


Figure 120 Description des contextes Architecture Système et Sûreté de fonctionnement

2.2. METHODE « BESOINS DE SYNCHRONISATION »

En appliquant la méthode « **Définir des besoins de synchronisation** », de nombreux besoins de synchronisation ont été identifiés. Ce chapitre n'en traitera que trois.

Dans un premier temps, une étude d'identification de potentielles interactions a été conduite en se focalisant sur les processus des disciplines d'ingénierie et les flux de données exploités ou produits. Le Figure 121 illustre les flux de données manipulés (gros grains) durant les activités des deux processus chez l'Equipementier A.

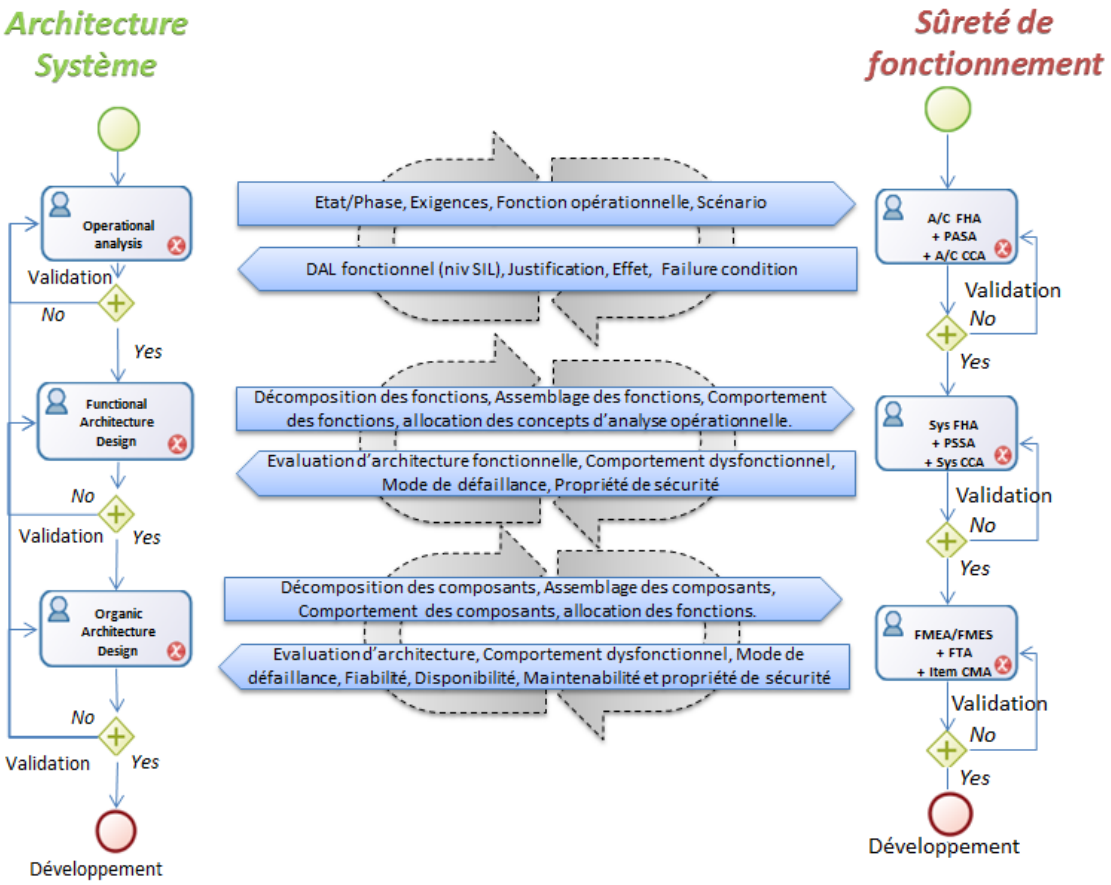


Figure 121 Identification de potentielles interactions selon les processus et les flux de données

Dans un second temps, une analyse croisée des activités des disciplines d'ingénierie et de leurs possibles dépendances a été conduite. Elle identifie les activités de chaque processus qui traite de préoccupations à niveau d'abstraction identique ou très proche. Ces résultats sont présentés dans la Table 34.

Note : Des travaux similaires ont également été mené par l'IRT Saint Exupéry dans le projet MOISE (cf. [138]). L'équipe propose notamment un processus de flux de données entre les activités des deux disciplines étudiés ici.

Table 34 Identification des points de synchronisation des activités des processus

Besoin de synchronisation entre des processus d'architecture système et de sûreté de fonctionnement		Analyse niveau avion		Analyse niveau système		Analyse niveau équipement		Analyse de cause commune (avion et système)		
		Aircraft FHA	PASA	System FHA	PSSA	System FTA	System FMEA/FMES	PRA	CMA	ZSA
Analyse opérationnelle	Définition de l'environnement du système et des cas d'utilisation	X	X	X				X		
	Définition des scénarios							X		
	Définition du cycle de vie du système	X		X						
Conception d'architecture fonctionnelle	Décomposition fonctionnelle du système		X	X	X	X	X	X		
	Architecture fonctionnelle			X	X	X	X	X	X	X
	Comportement fonctionnel des fonctions			X		X	X	X	X	X
Conception d'architecture organique	Décomposition organique					X	X	X	X	X
	Architecture organique					X	X	X	X	X
	Comportement fonctionnel des composants					X	X	X	X	X

Les 3 besoins de synchronisation (BS) ont été retenus de sorte à présenter des applications de synchronisation à trois niveaux d'abstraction différents dans les processus : l'un durant l'analyse opérationnelle, un autre durant l'analyse fonctionnelle et un dernier lors de l'analyse physique. Ils sont décrits dans la Table 35 et sont illustrés par la Figure 122 à l'aide du point de vue (définis au Chapitre II2.3.3).

Table 35 Définition des besoins de synchronisation

BS	**Besoin de l'architecture système vis-à-vis de la sûreté de fonctionnement**	**Besoin de la sûreté de fonctionnement vis-à-vis de l'architecture système**	**Raison de ce besoin**	**Points de vue concernés par cette interaction (2)**	**Les éléments, propriétés et/ou relations mise en jeu (2)**
BS1	*Vérifier la prise en compte de l'exhaustivité des cas d'utilisation sans omettre les cas indirects, transverses et/ou liés à la sécurité.*	*Tenir compte des cas d'utilisation en compléments des fonctions opérationnelles déjà identifiées dans les études de sûreté.*	*Mise en cohérence des cas d'utilisation*	*Diagramme de cas d'utilisation UML*	*UseCase, Subject, Include, Association, Extend, Actor.*
				Décomposition hiérarchique des fonctions opérationnelles	*Fonctions opérationnelles, relations de composition*
BS2	*Assurer un découpage sémantique et une décomposition fonctionnelle cohérents avec la sûreté de fonctionnement*	*Vérifier la cohérence des fonctions de l'architecte avec les évènements dysfonctionnels issus de la PSSA.*	*Mise en cohérence des fonctions*	*Diagramme de définition de blocks SysML*	*Block, Relation de composition, port*
				Modèle AltaRica 3.0 système fonctionnel	*Fonction, relations de composition, assertion*
BS3	*Présenter une solution d'architecture physique à la sûreté de fonctionnement. Permettre une évolution de la solution à partir des retours des disciplines.*	*Permettre d'évaluer la solution d'architecture physique à l'instant t proposée par l'architecte tout en garantissant une cohérence.*	*Mise en cohérence de l'architecture physique*	*Definition block diagram, Internal Block Diagram SysML*	*Block, Part, Connector, Port*
				Modèle AltaRica 3.0 système physique	*Block, Class, Assertion, Properties*

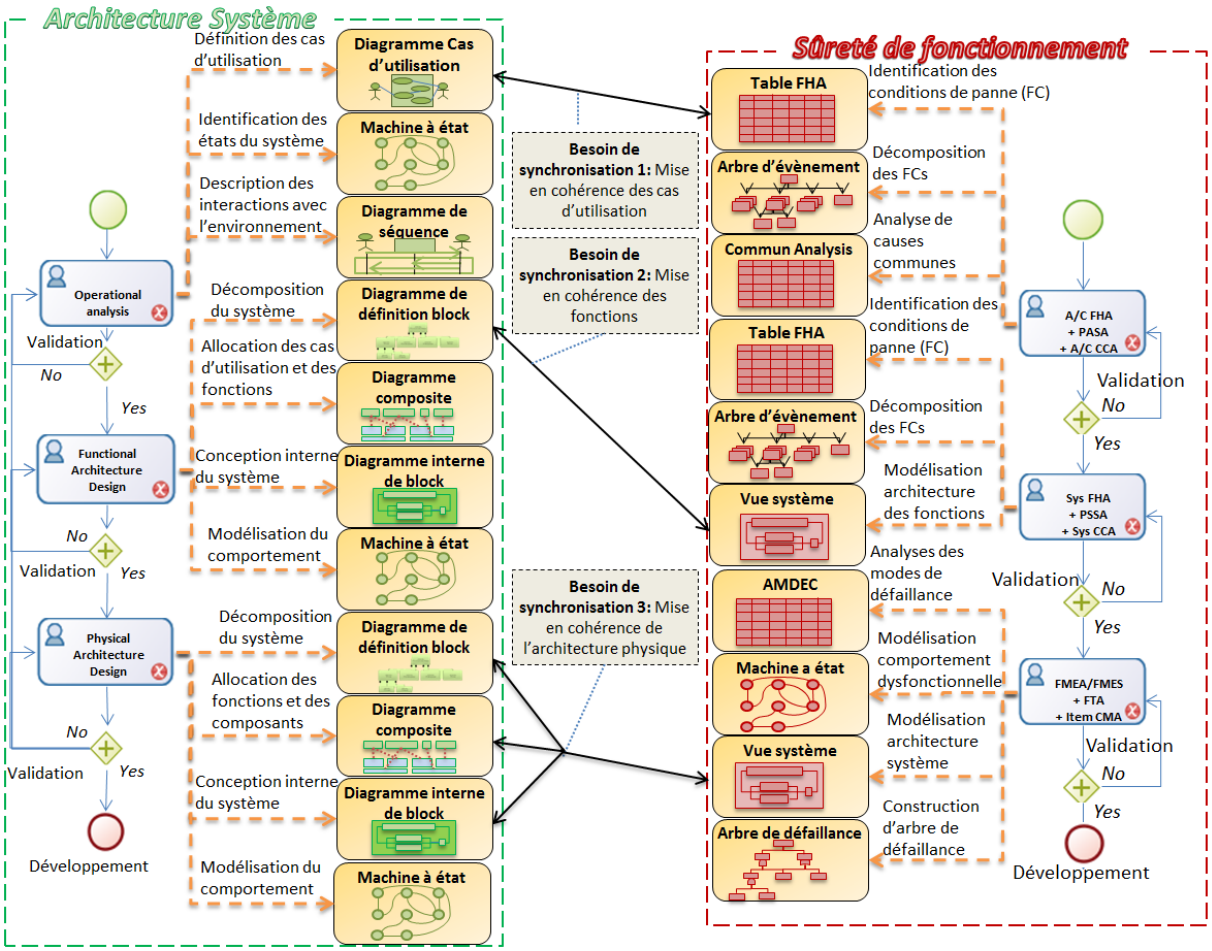


Figure 122 Besoins de synchronisation contextualisés

Le déroulement des processus permet de mener une réflexion sur l'ordonnancement des besoins de synchronisation. Il semble évident que BS1 devra être mis en cohérence avant de s'intéresser aux concepts liés aux fonctions BS2, car les fonctions sont construites à partir des concepts opérationnels. De même, l'architecture physique devra être mise en cohérence en dernier, car les composants du système sont choisis à partir des fonctions qu'ils remplissent. Ainsi l'ordonnancement logique, qui semble le plus adapté pour traiter les besoins de synchronisation chez l'Equipementier A, est :

$$Besoin\ de\ synchronisation\ n°1 < Besoin\ de\ synchronisation\ n°2$$
$$Et$$
$$Besoin\ de\ synchronisation\ n°2 < Besoin\ de\ synchronisation\ n°3$$

Le symbole "<" indique que l'argument de gauche doit être traité avant l'argument de droite.

3. MISE EN PRATIQUE DE LA CONFIGURATION DE SYNCHRONISATION

3.1. METHODE « FORMALISATION DES POINTS DE SYNCHRONISATION »

A partir des 3 BS, 5 points de synchronisation ont été définis par le responsable :

- **Point de synchronisation n°1** : Il traite de la mise en cohérence de la sémantique des cas d'utilisation du système (respectivement des fonctions opérationnelles). Il est issu du BS1.
- **Point de synchronisation n°2** : Il traite de la mise en cohérence de la sémantique et de la hiérarchie des fonctions. Il est issu du BS2.
- **Point de synchronisation n°3** : Il traite de la mise en cohérence de la hiérarchie des composants physiques. Il est issu du BS3.
- **Point de synchronisation n°4** : Il traite de la mise en cohérence des connexions des composants physiques. Il est issu du BS3.
- **Point de synchronisation n°5** : Il traite de la mise en cohérence de l'héritage des fonctions sur les composants physiques. Il est issu du BS3.

Le responsable de synchronisation a choisi de définir trois points de synchronisation à partir de BS3, car il souhaite séparer les concepts pivots d'architecture. Il fait l'hypothèse que les interactions seront facilitées entre les ingénieurs et les architectes dans cette configuration. Si cette séparation devenait trop coûteuse, il pourra la remettre en question au prochain cycle itératif de la méthodologie.

3.1.1. CONFIGURATION DU POINT DE SYNCHRONISATION N°1

La Figure 123 représente la configuration du **point de synchronisation 1**. Il intervient entre les points de vue « Diagramme de cas d'utilisation » en UML et « table hiérarchique » dans un tableur.

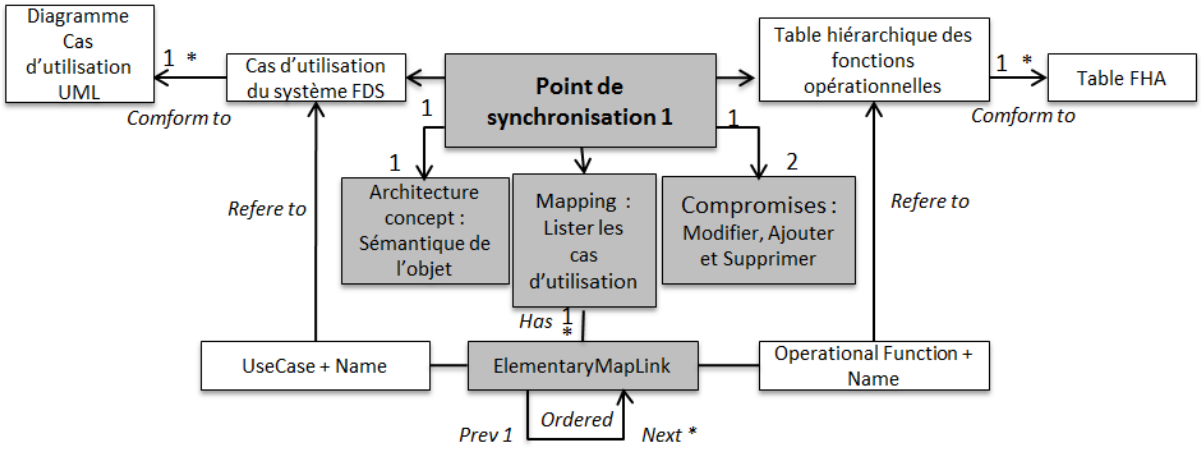


Figure 123 Configuration du point de synchronisation 1

Pour rappel, le concept pivot d'architecture (Cpa) est la combinaison des relations de structuration (R_{struc}) et un ensemble d'éléments (E_{type}) pour un objectif d'étude particulier ($ObjE$), cf. Chapitre II3.2.3. Le Cpa du point de synchronisation 1 est résumé dans la Table 36.

Table 36 Concept pivot d'architecture du point de synchronisation 1

Point de synchronisation 1 :	**Terme**	**Description**
	Cpa	Sémantique des éléments possédant un nom
	R_{struc}	Relation de composition
	E_{type}	« Abs_Usecase » comme élément parent et « PropertyName » comme élément fils.
	$ObjeE$	Mise en cohérence de la sémantique des cas d'utilisation du système.

Le métamodèle pivot correspondant au Cpa du point de synchronisation 1 est illustré Figure 124.

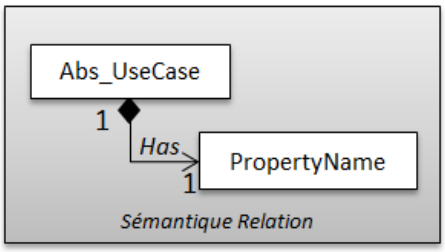


Figure 124 Métamodèle Cpa du point de synchronisation 1

Les mappings associés sont définis dans la Figure 125.

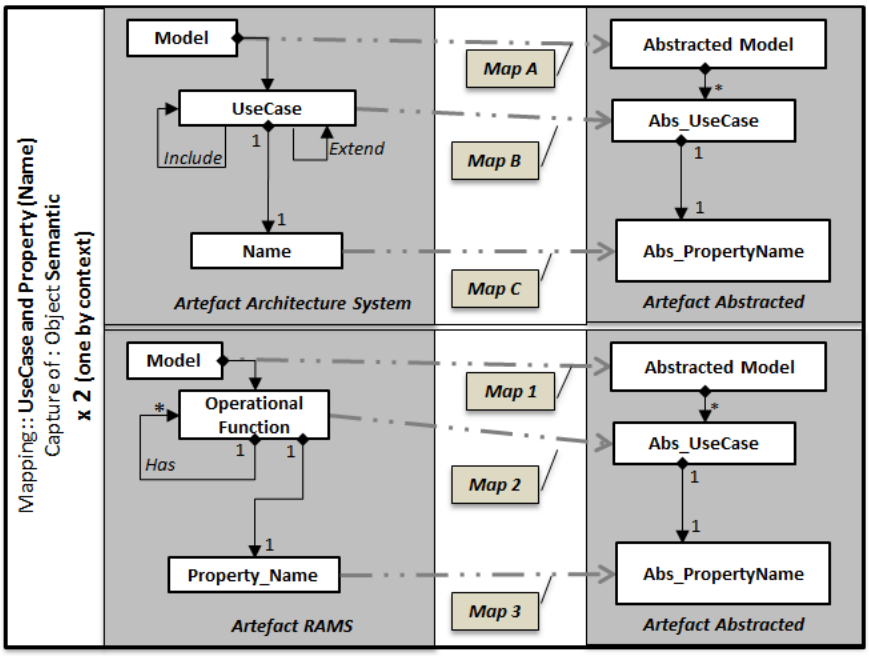


Figure 125 Mappings des abstractions du point de synchronisation 1

Les deux disciplines détiennent le même ensemble de compromis : « Ajouter un Abs_UseCase », « Supprimer un Abs_UseCase » ou « Renommer une PropertyName ».

3.1.2. CONFIGURATION DU POINT DE SYNCHRONISATION N°2

La Figure 126 représente la configuration du **point de synchronisation 2**. Il intervient entre les points de vue « Diagramme de définition de block » en SysML et « Structure des fonctions du système » spécifié en AltaRica 3.0 [70]. Ces points de vue semblent être redondants, cependant ils ne sont pas utilisés pour les mêmes objectifs d'étude.

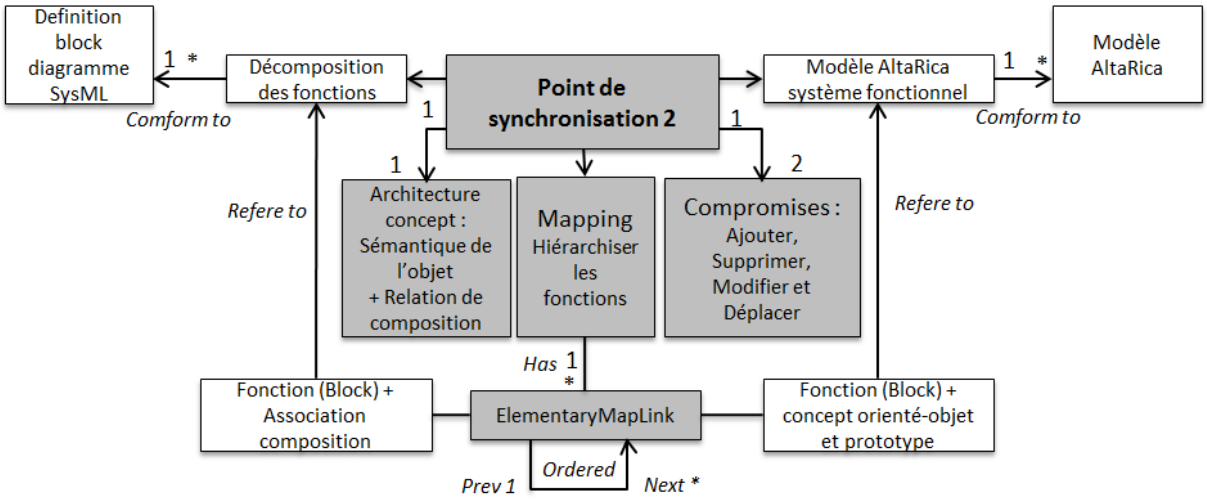


Figure 126 Configuration du point de synchronisation 2

Le Cpa du point de synchronisation 2 est résumé dans la Table 37.

Table 37 Concept pivot d'architecture du point de synchronisation 2

Point de synchronisation 2 :	**Terme**	**Description**
	Cpa	- Sémantique des fonctions possédant un nom - Composition des fonctions
	R_{struc}	Relation de composition (x2)
	E_{type}	- « Abs_Function » comme élément parent et « PropertyName » comme élément fils. - « Abs_Function » comme élément parent et « Abs_Function » comme élément fils.
	$ObjeE$	Mise en cohérence de la sémantique et de la composition des fonctions du système.

Le métamodèle pivot considéré pour le Cpa du point de synchronisation 2 est construit à partir de deux métamodèles élémentaires issus de la bibliothèque Cpa par défaut (cf. Chapitre II3.2.3). Il est représenté dans la Figure 127.

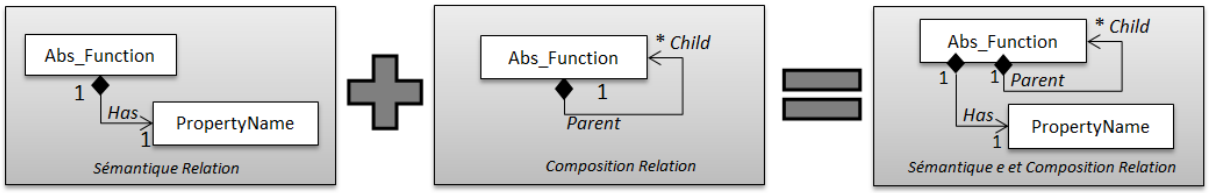


Figure 127 Métamodèle Cpa du point de synchronisation 2

Les mappings sont définis dans la Figure 128.

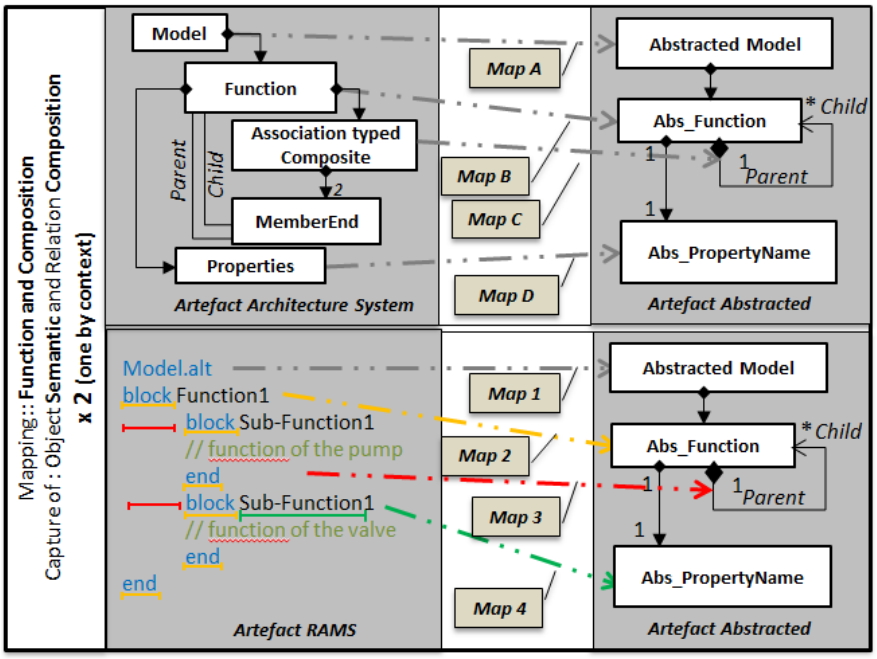


Figure 128 Mappings des abstractions du point de synchronisation n°2

Les deux disciplines détiennent le même ensemble de compromis : « Ajouter un Abs_Function », « Supprimer un Abs_Fonction », « Renommer une PropertyName » ou « Déplacer une Abs_Function ».

3.1.3. CONFIGURATION DU POINT DE SYNCHRONISATION N°3

La Figure 129 représente la configuration du **point de synchronisation 3**. Il intervient entre les points de vue « Diagramme de définition de block » en SysML et « Structure des composants du système » spécifié en AltaRica 3.0.

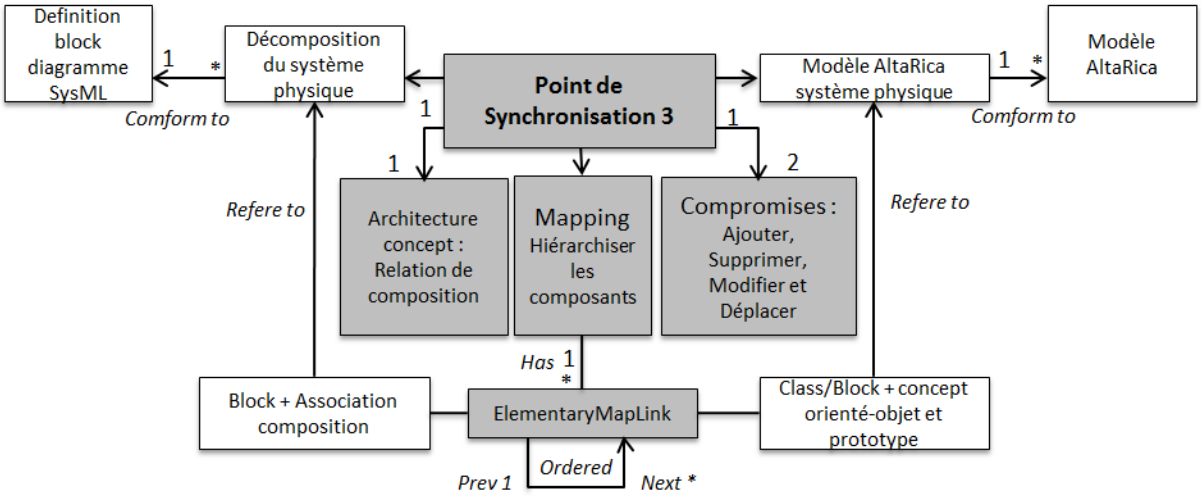


Figure 129 Configuration du point de synchronisation n°3

Le Cpa du point de synchronisation 3 est résumé dans la Table 38.

Table 38 Concept pivot d'architecture du point de synchronisation 3

Point de synchronisation 3 :	**Terme**	**Description**
	Cpa	Composition des composants du système
	R_{struc}	Relation de composition
	E_{type}	« Abs_Component » comme élément parent et « Abs_ Component » comme élément fils.
	$ObjeE$	Mise en cohérence de la composition des fonctions du système.

Le métamodèle pivot correspondant au Cpa du point de synchronisation 3 est illustré Figure 130.

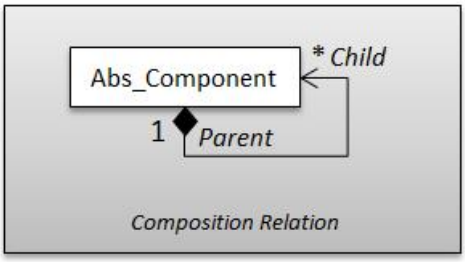


Figure 130 Métamodèle du Cpa du point de synchronisation 3

Les mappings sont définis dans la Figure 131.

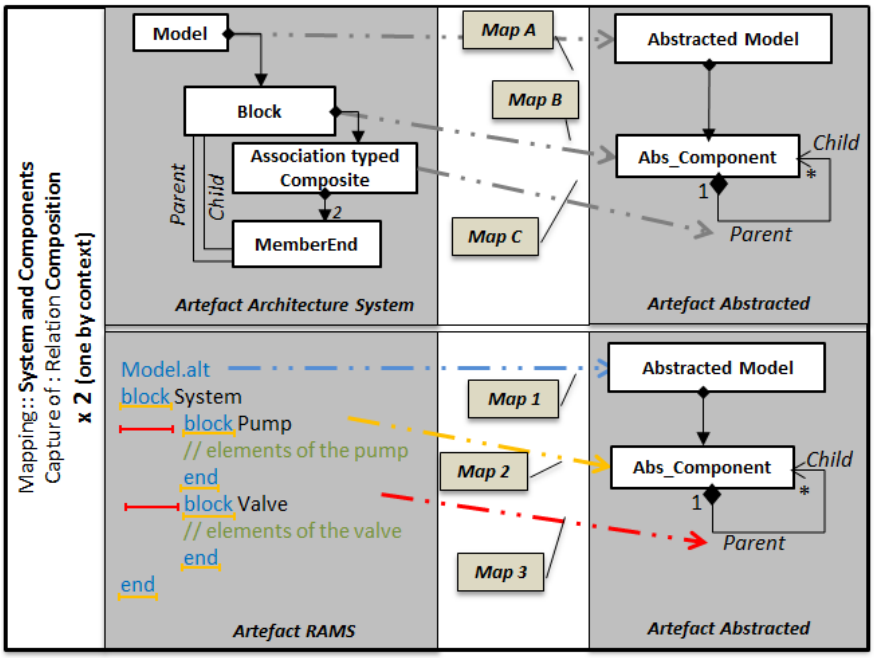


Figure 131 Mappings des abstractions du point de synchronisation n°3

Les deux disciplines détiennent le même ensemble de compromis : « Ajouter », « Supprimer », « Modifier » ou « Déplacer » une relation de composition.

3.1.4. CONFIGURATION DU POINT DE SYNCHRONISATION N°4

La Figure 132 représente la configuration du **point de synchronisation 4**. Il intervient entre les points de vue « Diagramme de blocks internes » en SysML et « Structure des composants du système » spécifié en AltaRica 3.0.

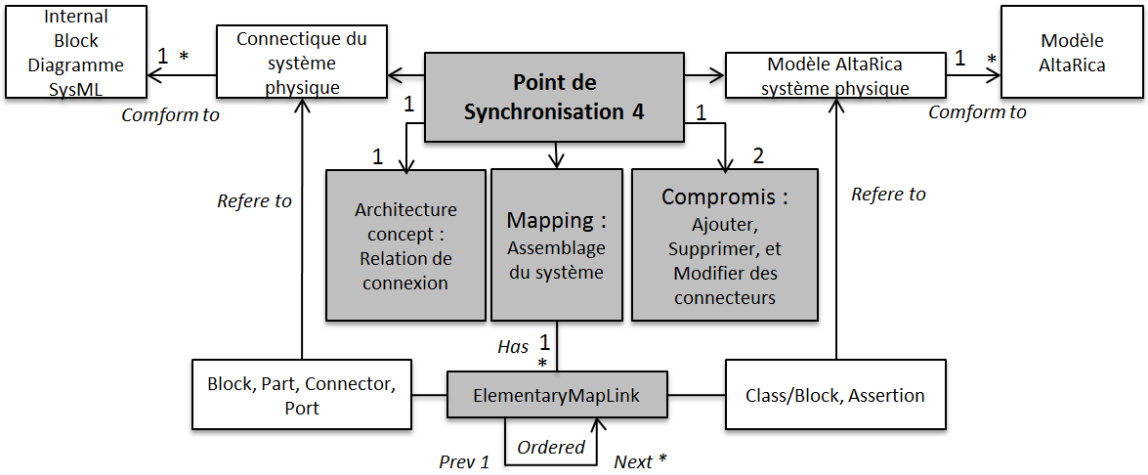


Figure 132 Configuration du point de synchronisation n°4

Le Cpa du point de synchronisation 4 est résumé dans la Table 39.

Table 39 Concept pivot d'architecture du point de synchronisation 4

Point de synchronisation 4 :	**Terme**	**Description**
	Cpa	Connexion des composants du système
	R_{struc}	Relation de connexion
	E_{type}	« Abs_Component » comme élément et « Connexion» comme relation de connexion.
	$ObjeE$	Mise en cohérence de la connexion des composants du système.

Le métamodèle pivot correspondant au Cpa du point de synchronisation 4 est illustré Figure 133.

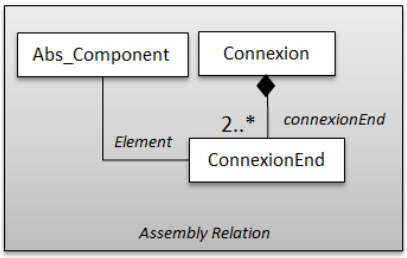


Figure 133 Métamodèle du Cpa du point de synchronisation 4

Les mappings sont définis dans la Figure 134.

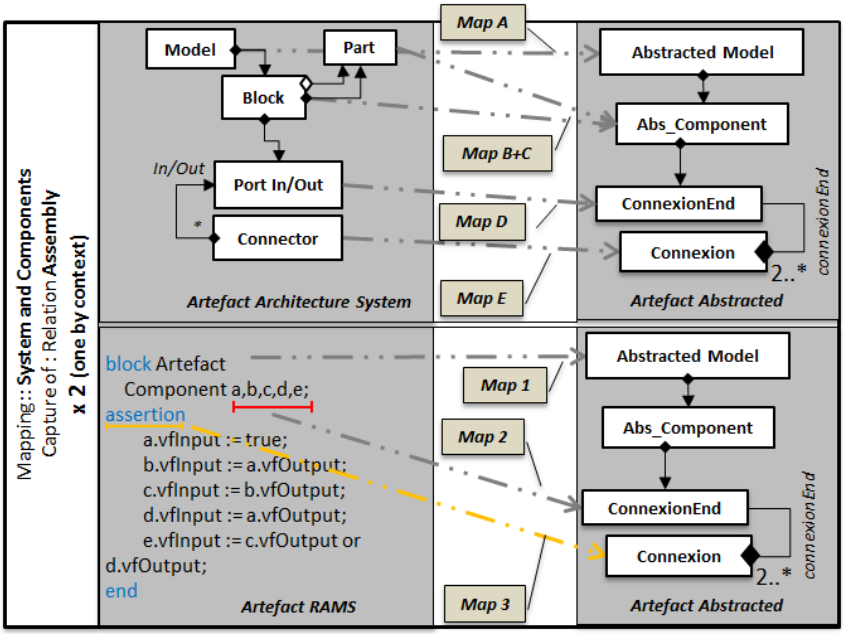


Figure 134 Mappings des abstractions du point de synchronisation n°4

Les deux disciplines détiennent le même ensemble de compromis : « Ajouter », « Supprimer » ou « Modifier » une relation de connexion.

3.1.5. CONFIGURATION DU POINT DE SYNCHRONISATION N°5

La Figure 135 représente la configuration du **point de synchronisation 5**. Il intervient entre les points de vue « Diagramme composite » en SysML et « Structure des composants du système » spécifié en AltaRica 3.0.

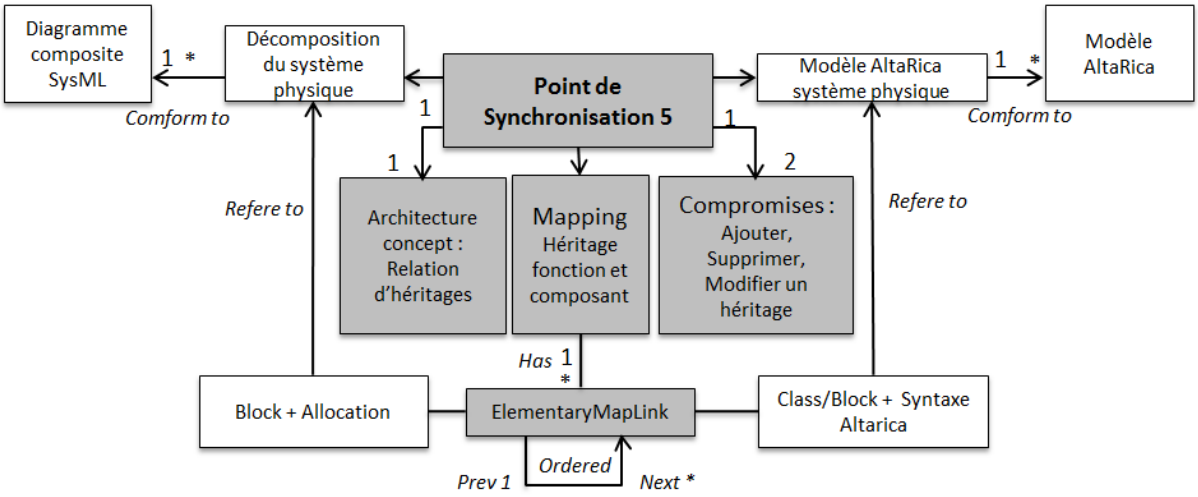


Figure 135 Configuration du point de synchronisation n°5

Le Cpa du point de synchronisation 5 est résumé dans la Table 40.

Table 40 Concept pivot d'architecture du point de synchronisation 5

Point de synchronisation 5	**Terme**	**Description**
	Cpa	Allocation des fonctions sur les composants du système
	R_{struc}	Relation d'héritage
	E_{type}	« Abs_Function » comme élément source et « Abs_Component » comme élément cible.
	$ObjeE$	Mise en cohérence des allocations des fonctions sur les composants du système.

Le métamodèle pivot correspondant au Cpa du point de synchronisation 5 est illustré Figure 136.

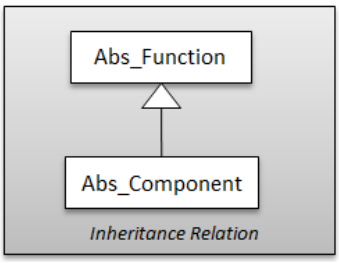


Figure 136 Métamodèle du Cpa du point de synchronisation 5

Les mappings sont définis dans la Figure 137

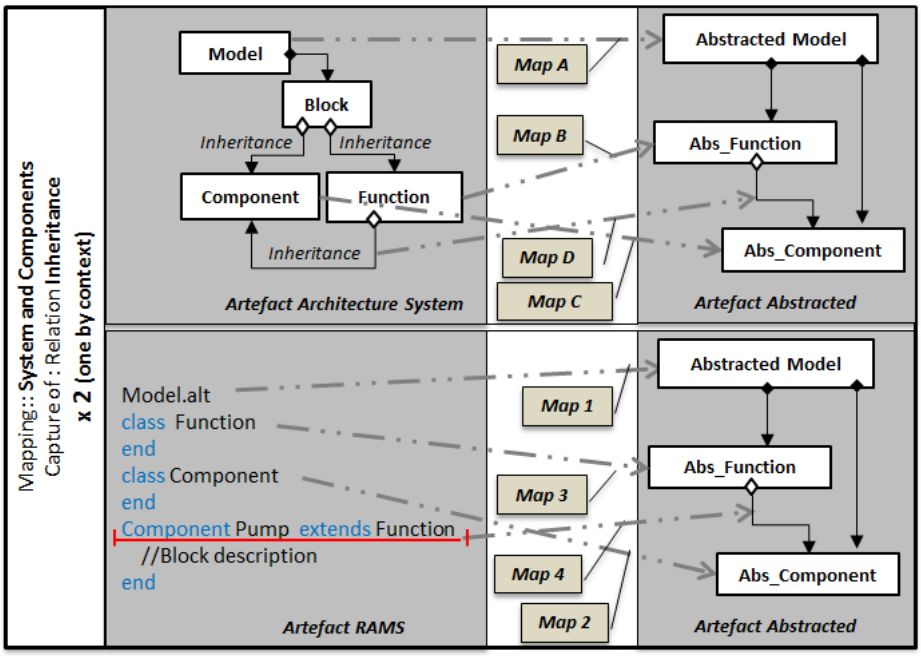


Figure 137 Mappings des abstractions du point de synchronisation n°5

Les deux disciplines détiennent le même ensemble de compromis : « Ajouter », « Supprimer » ou « Modifier » une relation d'allocation.

3.2. METHODE « ORDONNANCEMENT DES POINTS DE SYNCHRONISATION »

Une fois la configuration des points de synchronisation effectuée, le responsable peut définir les propriétés d'antériorité qui caractérisent l'ordre des exécutions.

Table 41 Ordonnancement des points de synchronisation

Point de synchronisation	**Antériorité**	**Point de vue Architecte système**	**Point de vue Sûreté de fonctionnement**
n°1	Aucun	Diagramme Cas d'utilisation UML	Table FHA
n°2	1	Définition block diagramme SysML	Modèle AltaRica 3.0
n°3	2	Définition block diagramme SysML	Modèle AltaRica 3.0
n°4	3	Diagramme de block interne SysML	Modèle AltaRica 3.0
n°5	2 et4	Diagramme de block interne SysML	Modèle AltaRica 3.0

Le résultat de la Table 41 peut être présenté sous la forme d'un processus de synchronisation de modèles en réutilisant les représentations des concepts définis au Chapitre II2.3.2. Dans la

Figure 138, trois lignes de vie sont construites : celles des disciplines d'ingénierie et celle des exécutions des points de synchronisation.

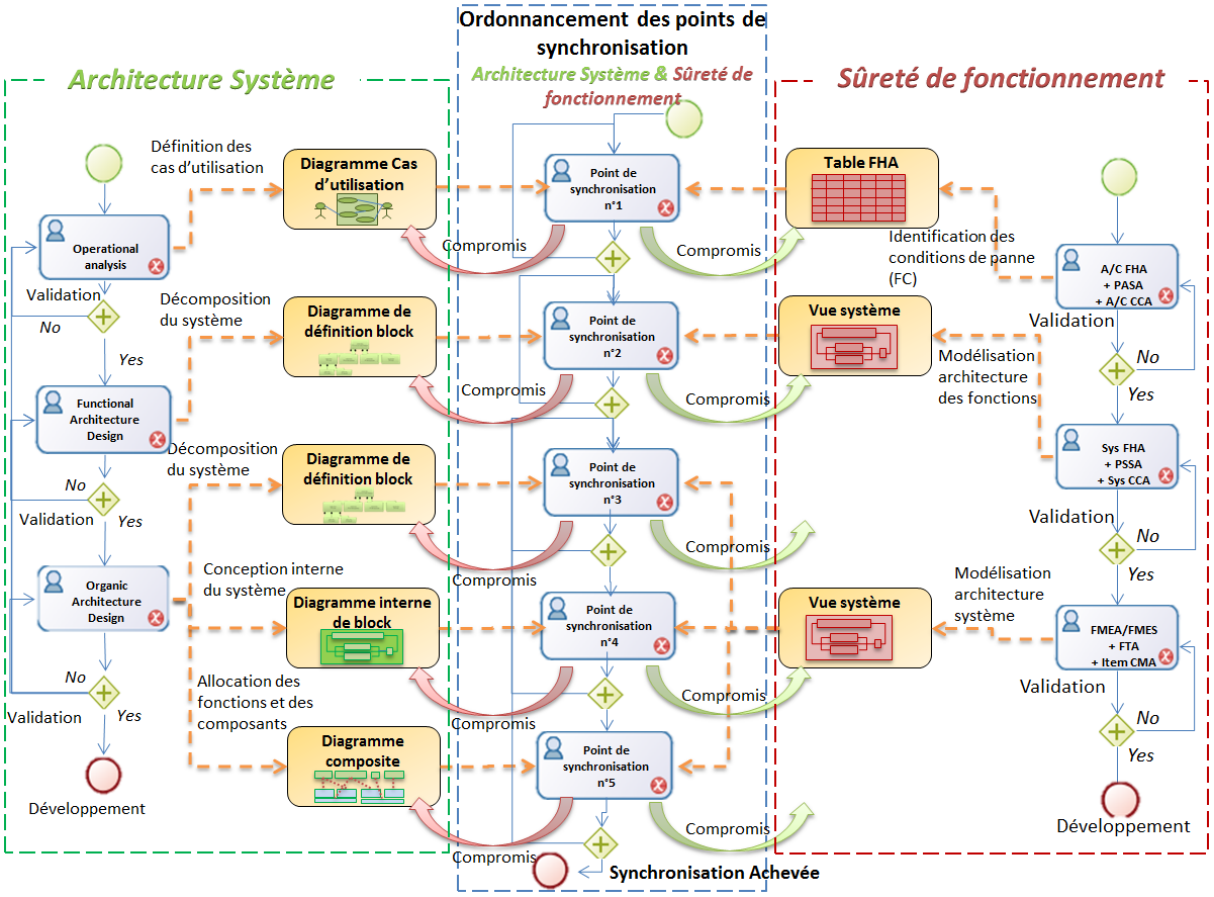


Figure 138 Ordonnancement des points de synchronisation contextualisées

Les points de synchronisation s'effectueront dans l'ordre **n°1 < n°2 < n°3 < n°4 < n°5**. Le symbole "<" indique que l'argument de gauche doit être effectué avant l'argument de droite.

4. MISE EN PRATIQUE DE L'APPLICATION DE LA SYNCHRONISATION

Toutes les vues sources utilisées pour les applications des points de synchronisation sont issues du cas d'étude présenté au Chapitre IV. Avant chaque application, le contexte du point de synchronisation est rappelé dans un tableau (Table 42, Table 43 et Table 44). Pour chaque point de synchronisation, une seule itération est déroulée complètement. Les itérations sont illustrées par une figure. Le contexte d'architecture système sera symbolisé en vert et le contexte de sûreté de fonctionnement en rouge.

Etant donné le nombre important d'éléments traités par les points de synchronisation 4 et 5, l'illustration de leurs applications est présentée en Annexes 2.4 et 2.6, tout comme les modèles de cohérence résultants, en Annexe 2.5.

4.1. APPLICATION DU POINT DE SYNCHRONISATION N°1

Table 42 Point de synchronisation n°1 contextualisé

Disciplines d'ingénierie	**Architecture système**	**Sûreté de fonctionnement**
Processus	ISO 15288 [35]	ARP 4754 [131]
Activités	Business or Mission Analysis Process (Clause 6.4.1)	Aircraft FHA
Méthode	Définition des cas d'utilisation	Identification des conditions de panne (FC)
Vues	Diagramme cas d'utilisation : **Figure 85 Vue Architecte Système – Analyse opérationnelle de l'hélicoptère (cf. p. 114)**	Vue hiérarchique des fonctions opérationnelles (Données d'entrée de la méthode) : **Table 21 Fonctions opérationnelles de l'hélicoptère pour la sûreté de fonctionnement (cf. p. 129)**
Eléments et Propriétés	UseCase, Include, Extend, Name, Model.	Operational Function, Model, Properties, Has.
Outil employés /Langages	Papyrus, SysML	MS Excel

La Figure 139 synthétise les étapes de la « méthode applicative de la synchronisation de modèles » (cf. Chapitre I5.2) sur le point de synchronisation 1.

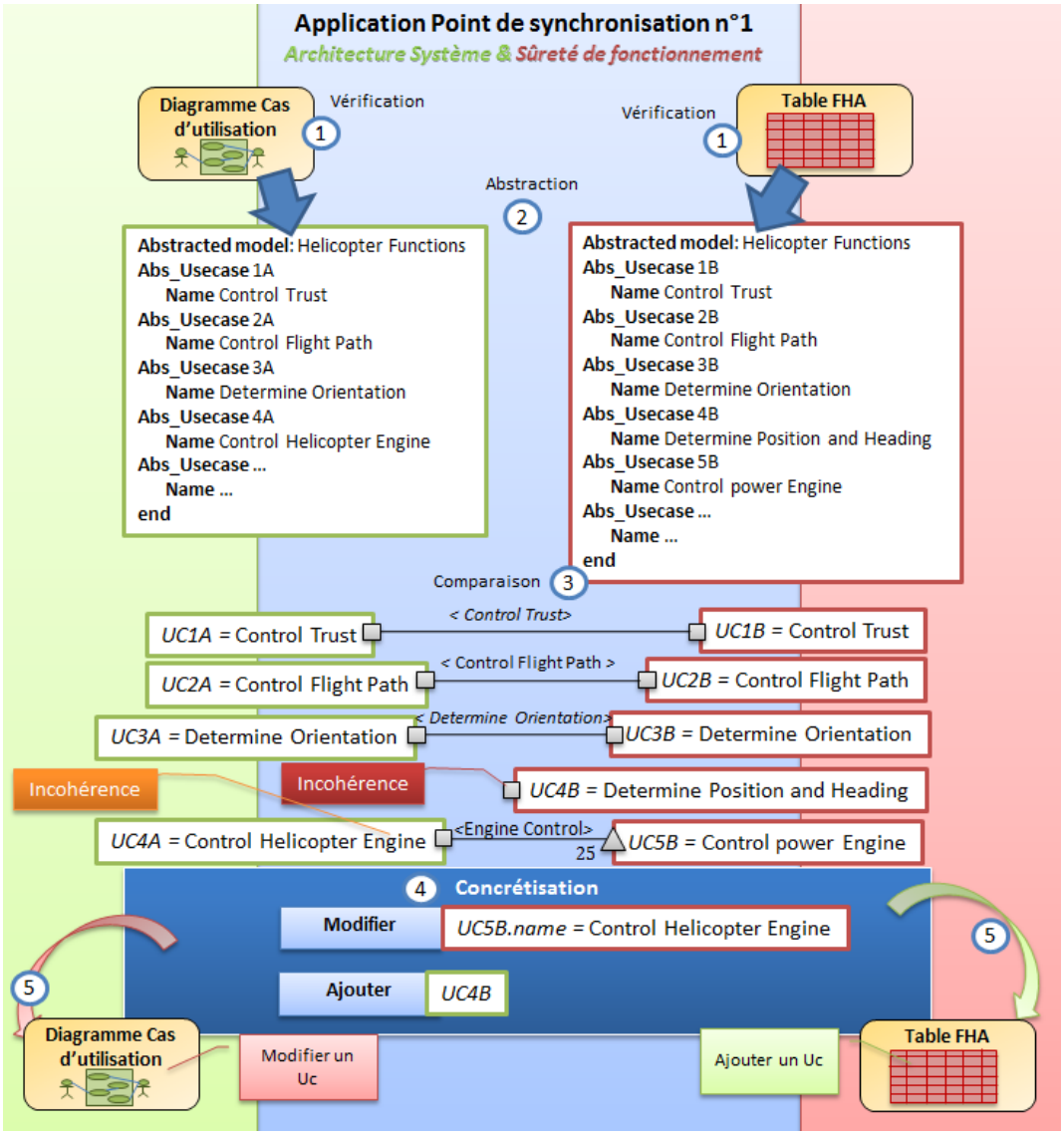


Figure 139 Application du point de synchronisation n°1

Deux incohérences sont identifiées :

- Il semble que le cas d'utilisation « Determine Position and Heading » soit manquant parmi la liste des cas d'utilisation de l'architecte système.
- Il n'y a pas de complétude suffisante entre « Control Helicopter Engine » et « Control power Engine ».

Les éléments de traces des abstractions, de la comparaison et des concrétisations sont alors générés.

4.2. APPLICATION DU POINT DE SYNCHRONISATION N°2

Avant d'appliquer la synchronisation sur le second cas, la Table 43 rappelle le contexte du point de synchronisation n°2.

Table 43 Point de synchronisation n°2 contextualisé

Disciplines d'ingénierie	**Architecte système**	**Sûreté de fonctionnement**
Processus	ISO 15288 [35]	ARP 4754 [131]
Activités	Stakeholder Needs & Requirements definition process (Clause 6.4.2)	Aircraft FHA
Méthode	Décomposition du système	Modélisation architecture des fonctions
Vues sources	Diagramme de définition block : **Figure 89 Décomposition fonctionnelle du système par un diagramme de définition des blocks SysML (cf. p. 122)**	Modèle AltaRica 3.0 : **Figure 107 Diagramme Fonctionnel du système de détection incendie (cf. p. 131),** **Figure 108 Modèle AltaRica Système Fonctionnel du système de détection incendie (cf. p. 132).**
Eléments et Propriétés	Model, Function, Association typed Composite, MemberEnd, Properties, Parent, Child	Model, Block / Class, Name, *Tabulation*
Outil employés /Langages	Papyrus, SysML	AltaRica 3.0

La Figure 140 Figure 139 synthétise les étapes de la « méthode applicative de la synchronisation de modèles » (cf. Chapitre I5.2) sur le point de synchronisation 2.

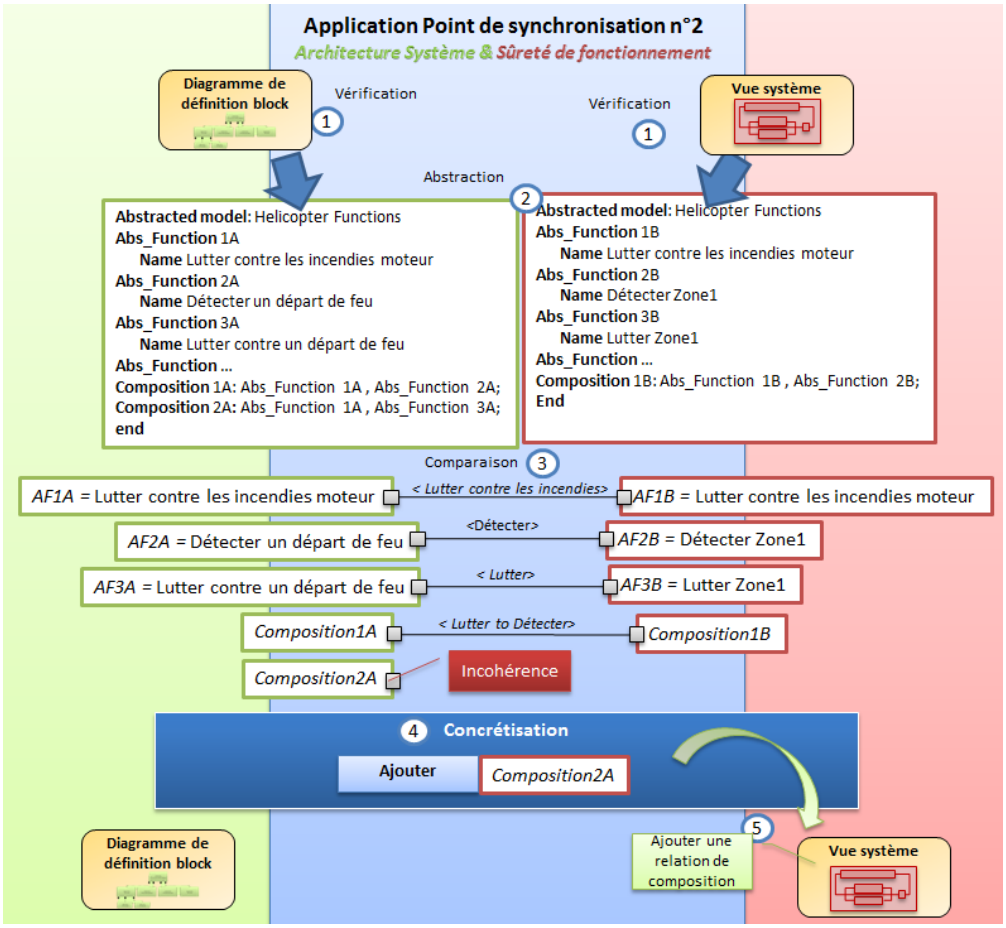


Figure 140 Application du point de synchronisation n°2

Une incohérence est identifiée : elle concerne la relation de composition entre les fonctions « Lutte contre les incendies moteur » et « Lutter contre un départ de feu ».

Les éléments de traces des abstractions, de la comparaison et des concrétisations sont alors générés.

4.3. APPLICATION DU POINT DE SYNCHRONISATION N°3

Avant d'appliquer la synchronisation, la Table 44 rappelle le contexte du point de synchronisation n°3.

Table 44 Point de synchronisation n°3 contextualisé

Disciplines d'ingénierie	**Architecture Système**	**Sûreté de fonctionnement**
Processus	ISO15288 [35]	ARP 4754 [131]
Activités	Architecture definition processes (Clause 6.4.4)	Aircraft FHA
Méthode	Décomposition du système	Modélisation de l'architecture système en vue d'analyse System CMA
Vues	Diagramme de définition block : **Figure 95 Décomposition de l'architecture physique (cf. p. 125).**	Modèle AltaRica 3.0 : **Figure 115 Modélisation en S2ML du système (cf. p. 139).**
Eléments et Propriétés	Model, Block, Association typed Composite, MemberEnd, Parent, Child.	Model, Block, Class, *name*, *tabulation.*
Outil employés /Langages	Papyrus, SysML	AltaRica 3.0

La Figure 141Figure 140 Figure 139synthétise les étapes de la « méthode applicative de la synchronisation de modèles » (cf. Chapitre I5.2) sur le point de synchronisation 3.

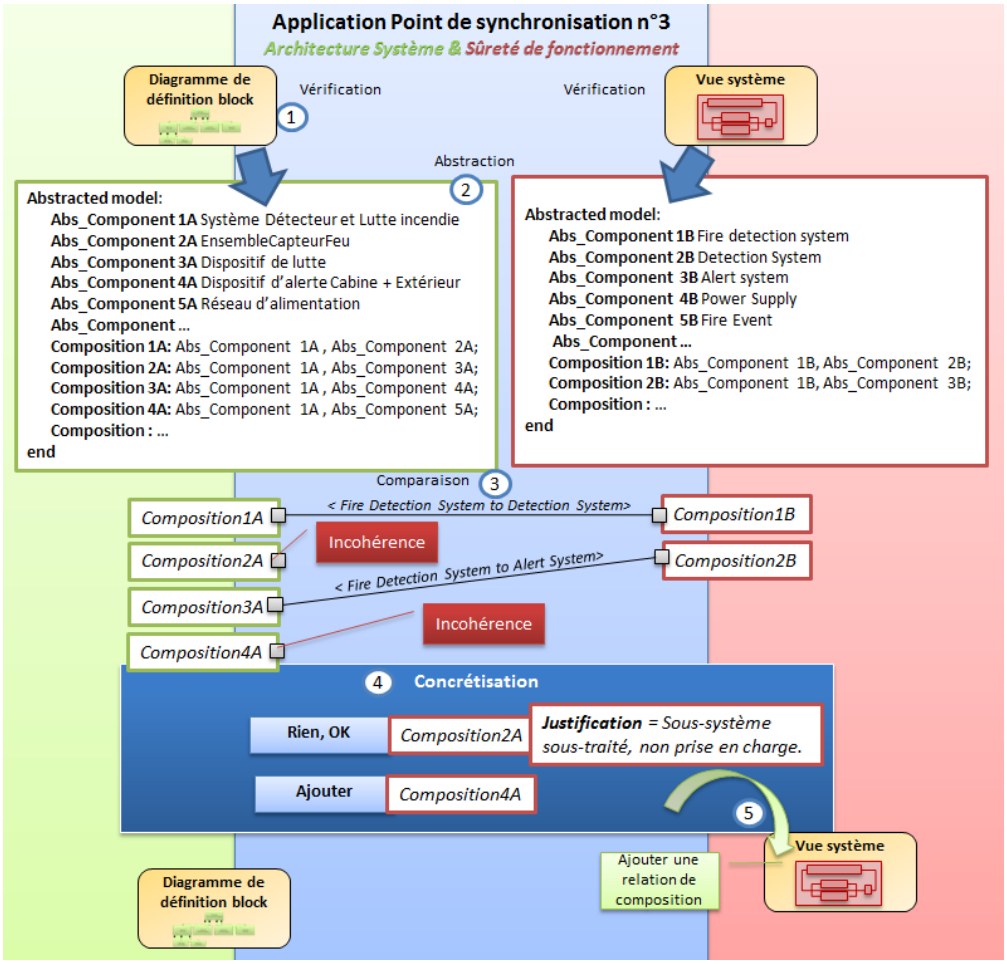


Figure 141 Application du point de synchronisation n°3

Deux incohérences sont identifiées :

- La première concerne la relation de composition entre les composants « Système de détection et Lutte incendie » et « Dispositif de lutte ».
- La seconde concerne la relation de composition entre les composants « Système de détection et Lutte incendie » et « Réseau d'alimentation »

Les éléments de traces des abstractions, de la comparaison et des concrétisations sont alors générés.

5. MISE EN PRATIQUE DU SUIVI ET EVOLUTION DES COHERENCES

L'application des 5 points de synchronisation a produit des traces pour les étapes d'abstraction, de comparaison et de concrétisation. Elles contiennent respectivement les relations de traçabilité issues : des exécutions des mappings, des résultats de comparaison et des exécutions des concrétisations.

On obtient ainsi un premier élément de trace (cf. Figure 142), « Traçabilité de la synchronisation » qui contient 5 éléments « Traçabilité d'un point de synchronisation » ainsi que « Traces des points de synchronisation » qui décrit l'ordre d'exécution des points de synchronisation.

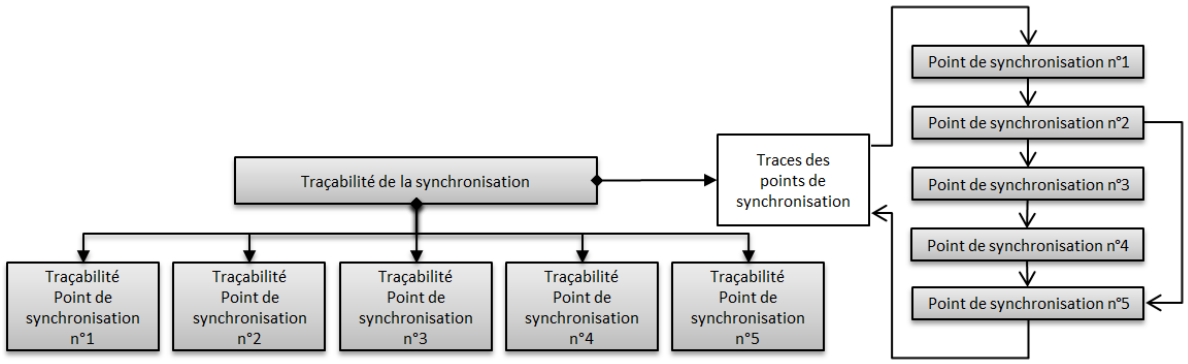


Figure 142 Traçabilité de l'application de la synchronisation

Pour illustrer, les traces du point de synchronisation n°1 sont présentées par les figures : Figure 143, Figure 144 et Figure 145.

Pour simplifier, on admettra que chaque point de synchronisation s'est terminé après 2 itérations. La « Traçabilité de l'itération 1 » (du point de synchronisation n°1) est représentée par les traces suivantes :

- Les traces des abstractions des deux modèles.

Traces d'abstraction : A => Abstraction de A
Liens de traçabilité :
UseCase Control Trust :=: **Abs_Usecase** 1A
Name Control Trust
UseCase Control Flight Path :=: **Abs_Usecase** 2A
Name Control Flight Path
UseCase Determine Orientation :=: **Abs_Usecase** 3A
Name Determine Orientation
UseCase Control Helicopter Engine :=: **Abs_Usecase** 4A
Name Control Helicopter Engine
UseCase ...:=: **Abs_Usecase** ...
Name ...
Model Diagramme Usecase **Model** Abstracted Model A
Metamodel A UML
Concept d'architecture Objet Sémantique

Traces d'abstraction : B => Abstraction de B
Liens de traçabilité :
Function Control Trust :=: **Abs_Usecase** 1B
Name Control Trust
Function Control Flight Path :=: **Abs_Usecase** 2B
Name Control Flight Path
Function Determine Orientation :=: **Abs_Usecase** 3B
Name Determine Orientation
Function Determine Position and Heading :=: **Abs_Usecase** 4B
Name Determine Position and Heading
Function Control power Engine :=: **Abs_Usecase** 5B
Name Control power Engine
Function ...:=: **Abs_Usecase** ...
Name ...
Model Table **Model** Abstracted Model B
Metamodel B Tableau hiérarchique
Concept d'architecture Objet Sémantique

Figure 143 Traces des abstractions des vues d'Architecture système et de Sûreté de fonctionnement

- La trace de la comparaison des deux modèles.

Comparaison de l'Abstraction de A et l'Abstraction de B
Ensemble de Cohérence:
Abs_Usecase 1A, 1 :=: **Abs_Usecase** 1B,1
Abs_Usecase 2A, 1 :=: **Abs_Usecase** 2B,1
Abs_Usecase 3A, 1 :=: **Abs_Usecase** 3B,1

Ensemble d'Incohérence:
Abs_Usecase 4A, 1 :=: **Abs_Usecase** 5B,0.25
Argumentation Null
Nature de l'incohérence Name

Abs_Usecase Null :=: **Abs_Usecase** 4B,1
Argumentation Null
Nature de l'incohérence Abs_Usecase

Model Abstracted Model A **Model** Abstracted Model B
Concept d'architecture Objet Sémantique

Figure 144 Traces de la comparaison des vues d'Architecture système et de Sûreté de fonctionnement

- Les traces des concrétisations des deux modèles et les compromis associés.

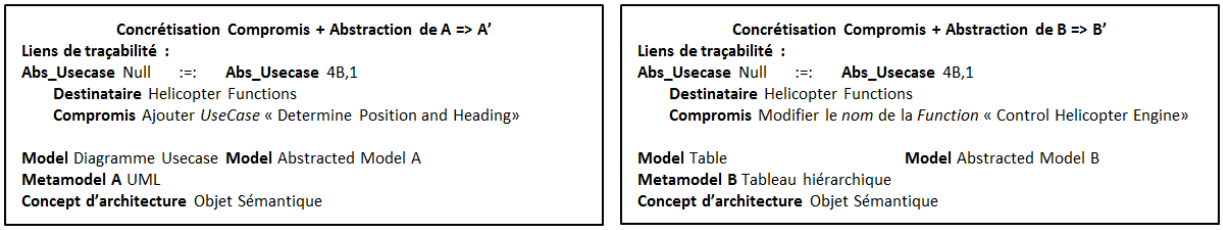


Figure 145 Traces des concrétisations des vues d'Architecture système et de Sûreté de fonctionnement

Les histoires qui se sont déroulées entre les disciplines d'ingénierie peuvent être ainsi explorées à plusieurs niveaux de raffinement (sur des vues processus, des vues élémentaires des modèles).

De nombreux résultats en ressortent :

- Les concepts pivots d'architecture les plus utilisés : le « concept d'objet sémantique » et le « concept de composition » ont été utilisés 2 fois chacun ;
- Les incohérences les plus identifiées et traitées : les incohérences liées aux relations d'héritage. Parmi les 27 incohérences, 13 traitent de relations d'héritage, 9 traitent de relations de connexion et 5 traitent de relation de composition ;
- Les incohérences qui n'ont jamais pu être traitées : beaucoup d'incohérences sont assumées par les ingénieurs ;
- Une amélioration de la cohérence des études dans le cycle de développement des projets.

Nous avons espoir que l'application d'une telle démarche permettra de constater les observations suivantes :

- L'esprit des ingénieurs s'est considérablement ouvert ;
- Le gain de temps se manifeste par l'efficacité des discussions entre les disciplines ;
- Toutes les activités sont tracées et permettent de voir l'évolution des synchronisations dans le temps ;
- La point de synchronisation en amont dans le processus permet une meilleure mise en cohérence et d'accompagner les ingénieurs dans leurs prises de décision dès que nécessaire.

La synchronisation de modèles ne définit pas de règle pour l'identification des relations de cohérence et des incohérences.

Pour vérifier l'efficacité de la synchronisation de modèles entre les disciplines, il est possible d'évaluer la réalisation des principes d'interactions définis dans la première étape de la méthodologie. D'autres questions peuvent être posées comme « Est-ce que l'action initialement souhaitée par le DG a été effectuée correctement ? Si non, peut-on faire une demande de changement ? » Les réponses peuvent amener à effectuer de nouvelles itérations de la méthodologie.

Chapitre VI IMPLEMENTATIONS ET EVALUATIONS

Les chapitres précédents présentaient un cadre conceptuel, une méthodologie générale et enfin un exemple d'application pour répondre aux problématiques d'interactions multidisciplinaires à l'aide de la synchronisation de modèles. Ces travaux ont nécessité des développements informatiques, ces derniers sont présentés dans ce chapitre.

Ce chapitre détaille les principaux développements effectués durant la thèse. Il vient démontrer la faisabilité technologique de la synchronisation de modèles au travers de l'abstraction, la comparaison et la concrétisation. Les travaux d'implémentation sont notamment :

- Le développement de points de vue et d'un profil SysML [2] ;
- Les algorithmes de comparaison ;
- Les transformations de modèles, abstractions et concrétisations de modèles.

1. VUES ET POINTS DE VUE

Une vue d'architecture se caractérise par un point de vue d'architecture de la même façon qu'un modèle se caractérise par un métamodèle (selon l'ISO 42010 [8]). La Figure 146 présente la dépendance des concepts de point de vue d'architecture et de vue d'architecture.

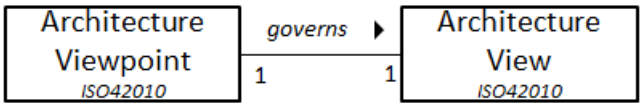


Figure 146 Dépendance des concepts Architecture Viewpoint et Architecture View selon ISO 42010

Plusieurs techniques d'implémentation permettent de définir des points de vue dans un environnement de modélisation. Ces techniques varient selon l'environnement et les langages implémentés. Elles peuvent utiliser des mécanismes d'extension tels que les « profils » (avec UML ou SysML) ou les « Error Annex » (avec AADL ou EAST-ADL), ou encore définir un DSML. Ces notions ont été présentées au Chapitre I2.7.2 et au Chapitre I4.1.

Cinq points de vue ont été définis dans cette section. Ils sont présentés indépendamment des outils qui les implémentent à l'exception d'un seul qui a été implémenté à l'aide d'un profil sur Papyrus [134].

- Point de vue « Contexte d'une discipline d'ingénierie » ;
- Point de vue « Contextualisation des besoins de synchronisation » ;
- Profil « Définition des contextes d'ingénierie et des besoins de synchronisation » ;
- Point de vue « Mapping » d'un point de synchronisation ;
- Point de vue « Contextualisation des points de synchronisation » ;
- Point de vue « Fonction de transformation ».

1.1. POINT DE VUE CONTEXTE D'UNE DISCIPLINE D'INGENIERIE

Lors de la formalisation des « contextes de disciplines d'ingénierie », un point de vue a été utilisé pour définir les activités, les méthodes et les vues employées par une discipline. Il est associé au métamodèle présenté à la Figure 14 (cf. Chapitre II2.3.2). Des exemples de vues ont aussi été présentés à la Figure 15.

Le point de vue « Contexte d'une discipline d'ingénierie » est illustré à la Figure 147.

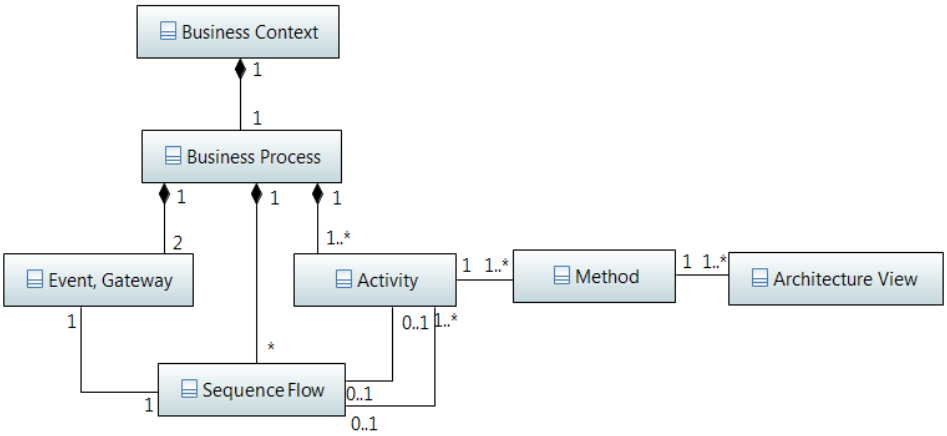


Figure 147 Point de vue « Contexte d'une discipline d'ingénierie »

Le point de vue se concrétise dans une vue par des éléments et des relations associées à une représentation graphique (syntaxe graphique). Les associations entre les concepts et leurs représentations graphiques sont représentées dans la Table 45. Un exemple de vue respectant le point de vue est présenté dans la Figure 148.

Table 45 Table d'association des concepts et la syntaxe graphique associée au point de vue "contexte d'une discipline d'ingénierie"

Concept du métamodèle correspondant	*Syntaxe graphique des éléments de la vue*	*Concept du métamodèle correspondant*	*Syntaxe graphique des éléments de la vue*
Business context	Nom	*Method*	Méthode nom
Business Process	Activité i → Activité i+1	*Architecture View*	Vue nom
Activity	Activité nom	*Event, Gateway, etc.*	
Sequence Flow			

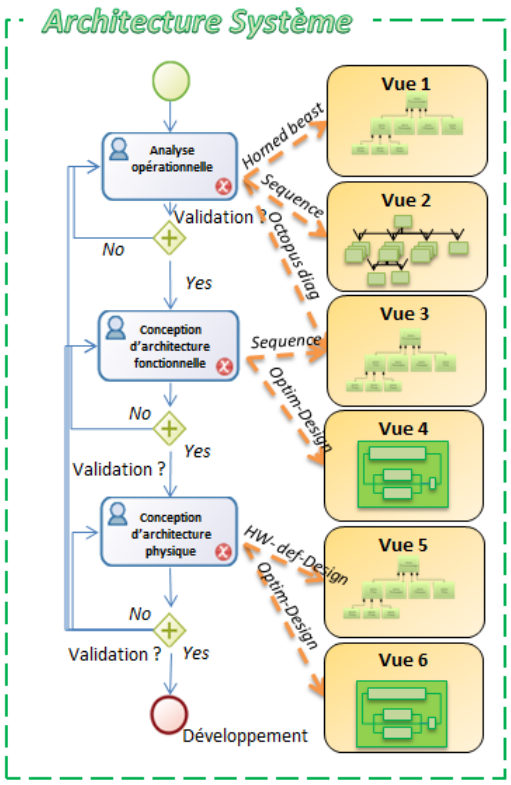


Figure 148 Exemple d'une vue du "Contexte d'une discipline d'ingénierie"

1.2. POINT DE VUE CONTEXTUALISATION DES BESOINS DE SYNCHRONISATION

Le second point de vue complète le précédant. A la définition et à la syntaxe graphique est ajouté le concept de besoin de synchronisation. Le point de vue « contextualisation des besoins de synchronisation » a été présenté au Chapitre II2.3.3.

Le point de vue « Contextualisation des besoins de synchronisation » est présenté Figure 149.

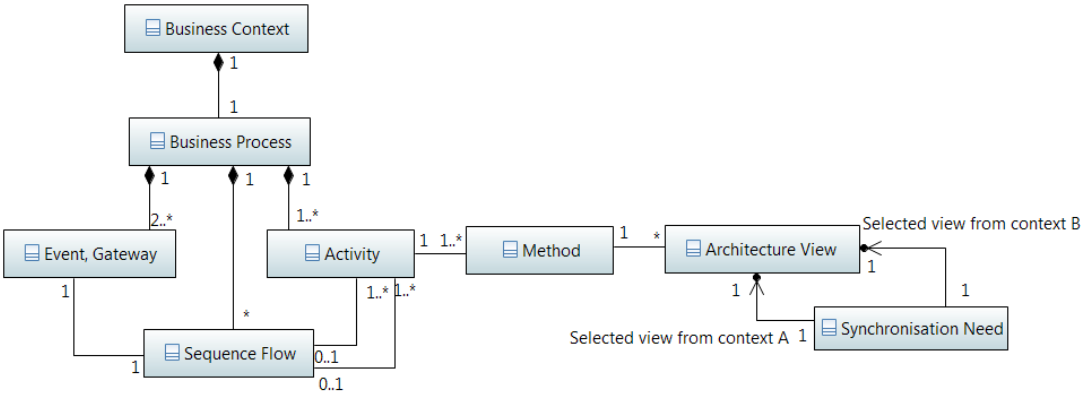


Figure 149 Point de vue «Contextualisation des besoins de synchronisation»

Ce point de vue positionne les contextes de deux disciplines d'ingénierie (respectivement A et B) en vis-à-vis et permet de tracer des relations caractérisant le besoin de synchronisation entre les vues d'architectures (Architecture View). Il permet de contextualiser le besoin et d'avoir un premier aperçu du déroulement des besoins retenus.

La Table 46 présente l'association entre le concept de besoin de synchronisation et la syntaxe graphique.

Table 46 Table d'association des concepts et la syntaxe graphique associée au point de vue "contextualisation des besoins de synchronisation"

Concept du métamodèle correspondant	***Syntaxe graphique des éléments de la vue***
Ensemble des concepts du point de vue «Contexte d'une discipline d'ingénierie ».	Cf. syntaxe graphique de la table d'équivalence de la vue «Contexte d'une discipline d'ingénierie».
Synchronisation Need	Besoin de synchronisation : Vue i *nom* <nom> Vue j *nom*

Des exemples de vue associés à ce point de vue sont utilisés au Chapitre II2.3.3, au Chapitre I3.3 et au Chapitre V2.2.

1.3. PROFIL UML DE DEFINITION DES CONTEXTES D'INGENIERIE

Les points de vue précédents sont particulièrement utiles lors des travaux pour décrire les disciplines d'ingénierie et les besoins de synchronisation. Ils offrent une vision globale des activités et des pratiques des disciplines. Un profil a été construit sur l'outil de modélisation Papyrus en UML [134]. Il pourra être employé pour outiller une partie de la méthodologie et permettre une démarche dite '*model-based*'.

Pour cela on utilise un profil qui permet d'étendre une *Metaclass*, i.e. élément d'un métamodèle, avec un stéréotype. Il permet d'étendre la sémantique d'un métamodèle avec des propriétés plus spécifiques. Dans notre cas, la sémantique désirée est déjà définie. Même si, les concepts n'ont pas de représentation graphique, le profil est utilisé de manière détourné pour exploiter les représentations graphiques des *Metaclass* d'UML. Le profil « Contextualisation des besoins de synchronisation » est décrit suivant la Figure 150.

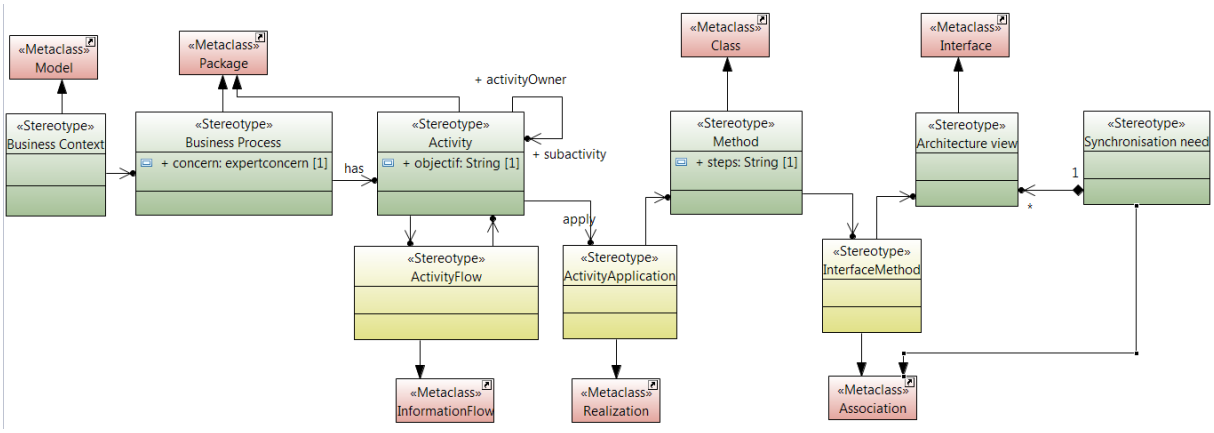


Figure 150 Profil UML « Contextualisation et besoin de synchronisation – Spécification »

L'un des principaux inconvénients de cette approche est que les *metaclass* choisies doivent avoir une syntaxe graphique. Ainsi, le métamodèle résultant considère certaines caractéristiques d'UML qui ne sont pas toujours souhaitées. Il est possible de limiter la sémantique du métamodèle en ajoutant des contraintes OCL, [39] au profil.

Le profil permet alors de représenter les éléments de notre vue à partir des représentations des éléments du langage UML, tel que présenté Table 47.

Table 47 Association du point de vue "Contextualisation des besoins de synchronisation" aux éléments de la Metaclass UML

Concept du point de vue correspondant	***Element Metaclass UML utilisée pour le profil***	***Concept du point de vue correspondant***	***Element Metaclass UML utilisée pour le profil***
Business Context	« Metaclass » Model::UML	*Method*	« Metaclass » Realization::UML
Business Process	« Metaclass » Package::UML	*Architecture View*	« Metaclass » Class::UML
Activity	« Metaclass » Package::UML	*Event, Gateway*	« Metaclass » Association::UML
Sequence Flow	« Metaclass » InformationFlow::UML	*Synchronisation Need*	« Metaclass » Association::UML

La Figure 151 et la Figure 152 sont deux exemples d'application du profil sur Papyrus. Ils permettent de représenter les vues : « contexte d'une discipline d'ingénierie » et « contextualisation des besoins de synchronisation ». Ces modèles sont présentés à titre illustratif, ils n'ont pas vocation à être communiqués mais plutôt à définir les contextes des disciplines d'ingénierie le plus explicitement possible.

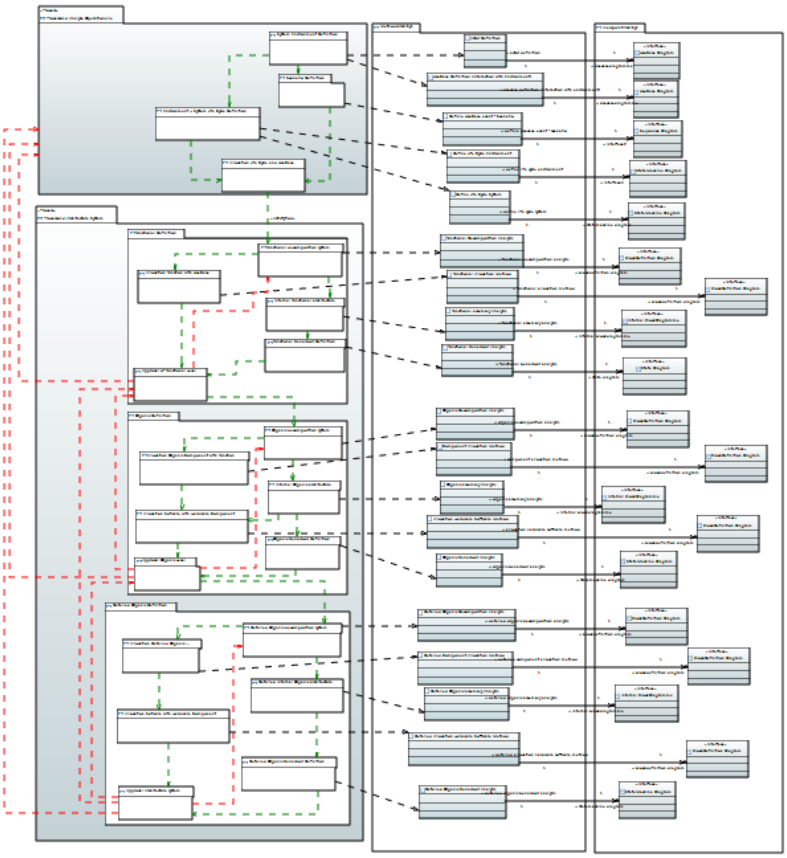


Figure 151 Exemple de modèle de disciplines d'ingénierie avec le profil "Contextualisation et besoin de synchronisation"

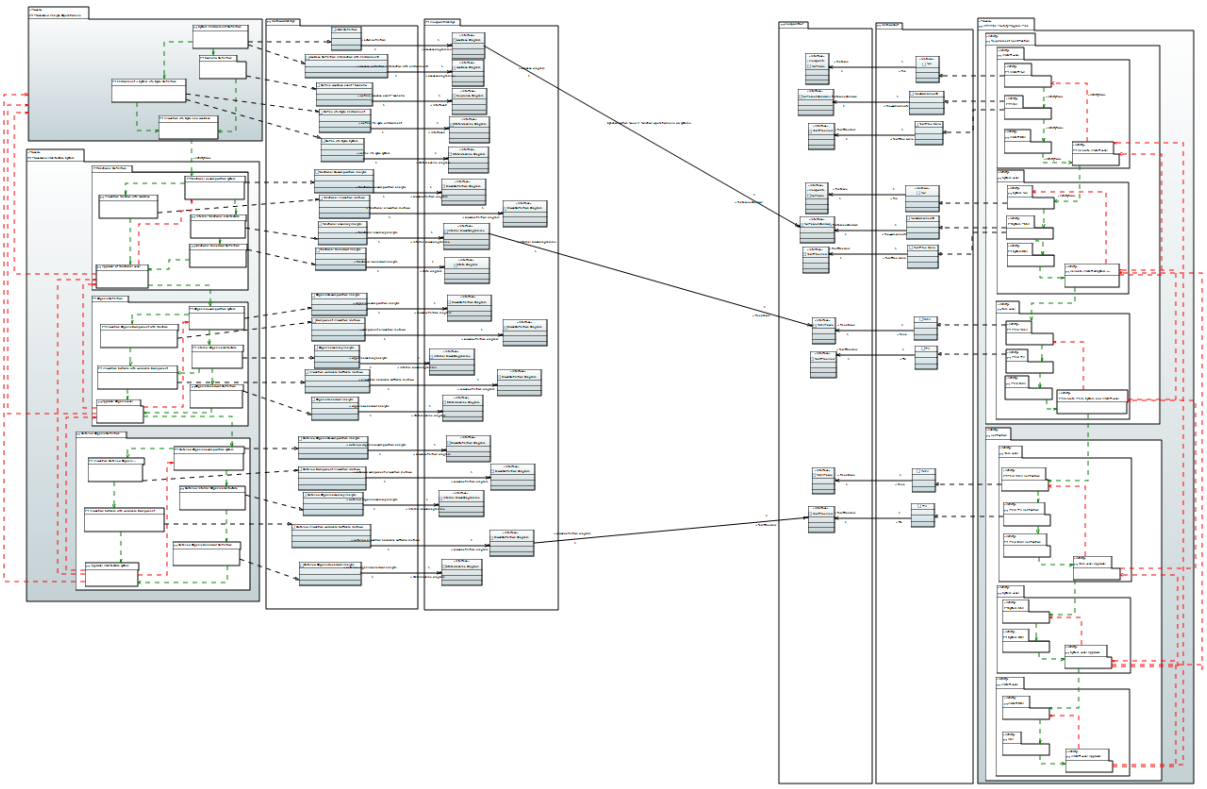


Figure 152 Exemple de modèle de contextualisation de besoins de synchronisation avec le profil "Contextualisation et besoin de synchronisation"

1.4. POINT DE VUE DES MAPPINGS D'UN POINT DE SYNCHRONISATION

Le point de vue « Mappings d'un point de synchronisation » est défini pour spécifier l'exécution des abstractions et des concrétisations. Il est spécifique aux transformations de modèles (cf. Chapitre II6.3.1). La Figure 153 présente le métamodèle du concept « Mapping ».

Il permet de relier des concepts et les relations des points de vue source et cible par des relations « Transformation map » contenant chacune une fonction de transformation.

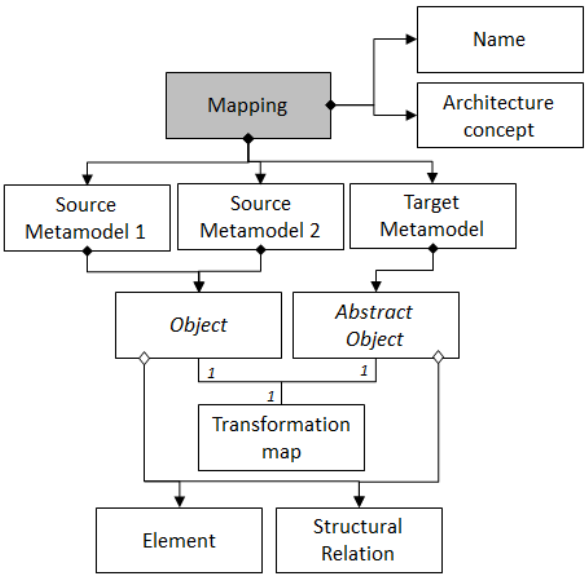


Figure 153 Métamodéle de « Mapping » dans la synchronisation de modèles

La Table 48 présente la syntaxe graphique des concepts du point de vue « Mapping ».

Table 48 Equivalence des concepts sémantiques avec la syntaxe graphique du « Mapping »

Concept du métamodèle correspondant	Syntaxe graphique des éléments de la vue	Concept du métamodèle correspondant	Syntaxe graphique des éléments de la vue
Name	Mapping:: <Name>	Relation Composition	Element → Element Contenu dans une région verte ou rouge.
Architecture concept	Capture of : <Architecture concept>	Relation Héritage	Element ◇→ Element Contenu dans une région verte ou rouge.
Source Metamodel 1	Artefact RAMS	Relation Association	Element — Element Contenu dans une région verte ou rouge.
Source Metamodel 2	Artefact Architecture System	Source Metamodel 1, Source Metamodel 2	« Model textuel » Contenu dans une région verte.
Target Metamodel	Artefact Abstracted	Map relation area	
Element	Element Contenu dans une région verte ou rouge.	Transformation Map	Element ··· Element Map i Contenu dans une région bleu.

Les concepts « Source Metamodel 1 », « Source Metamodel 2 », « Target Metamodel » sont abstraits, i.e. leurs syntaxes ne sont pas nécessairement représentées dans les vues. Elles sont là pour contraindre la « mise en page » de la vue du Template Figure 154. Ce sont des régions dans lesquelles certains éléments doivent être représentés.

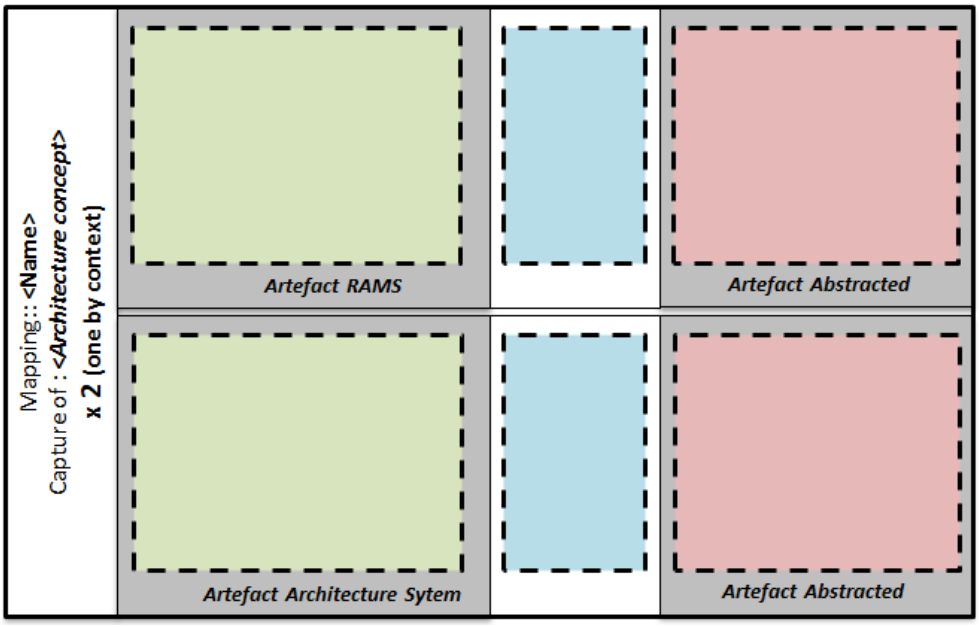


Figure 154 Template du point de vue des Mappings

De nombreux exemples respectant ce point de vue et sa syntaxe sont utilisés dans le mémoire : Chapitre I4.3 et Chapitre V3.1.

1.5. POINT DE VUE CONTEXTUALISATION DES POINTS DE SYNCHRONISATION

Le point de vue concerne « l'ordonnancement des points de synchronisation ». Il représente le processus de synchronisation construit durant l'activité de la méthodologie : « Application du point de synchronisation n° i ».

Le point de vue, le métamodèle associé, ainsi que des exemples de vues, sont employés au Chapitre II4.2.3, au Chapitre I4.3 et au Chapitre V3.2. Le point de vue « Contextualisation des besoins de synchronisation » est illustré dans la Figure 155.

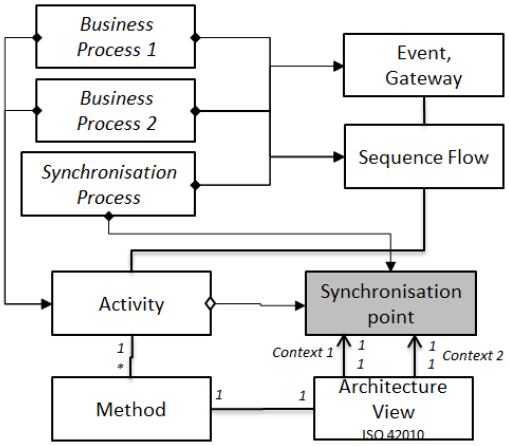


Figure 155 Métamodéle de « Point de synchronisation » dans la synchronisation de modèles

A chaque élément du point de vue est attribuée une syntaxe graphique (élément graphique). Ces associations sont représentées dans la Table 49.

Table 49 Associations des concepts avec la syntaxe graphique du « Point de synchronisation »

Concept du métamodèle correspondant	Syntaxe graphique des éléments de la vue	Concept du métamodèle correspondant	Syntaxe graphique des éléments de la vue
Business Process		Event, Gateway	
Synchronisation Process		Method	Contenu dans une région verte ou rouge.
Activity	Activity Contenu dans une région verte ou rouge.	Architecture View	View name Contenu dans une région verte ou rouge.
Synchronisation point	Point de synchronisation n° Contenu dans une région bleu.	Association Architecture View -> Synchronisation Point Context 1	View i name Point de synchronisation n° Interface entre une région verte et bleu.
Sequence Flow	Activity → Activity	Association Architecture View -> Synchronisation Point Context 2	Point de synchronisation n° View j name Interface entre une région rouge et bleu.

Un template associé au point de vue, Figure 156, définit également les régions verte, bleu et rouge dans lesquelles certains concepts sont contraints.

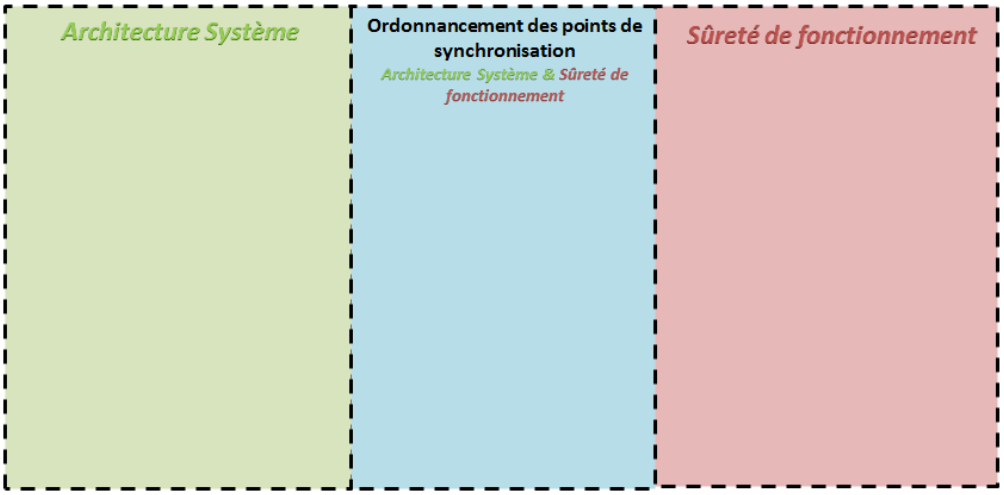


Figure 156 Template du point de vue des ordonnancements des points de synchronisation

Une vue respectant la syntaxe graphique est construite dans un exemple, Figure 157.

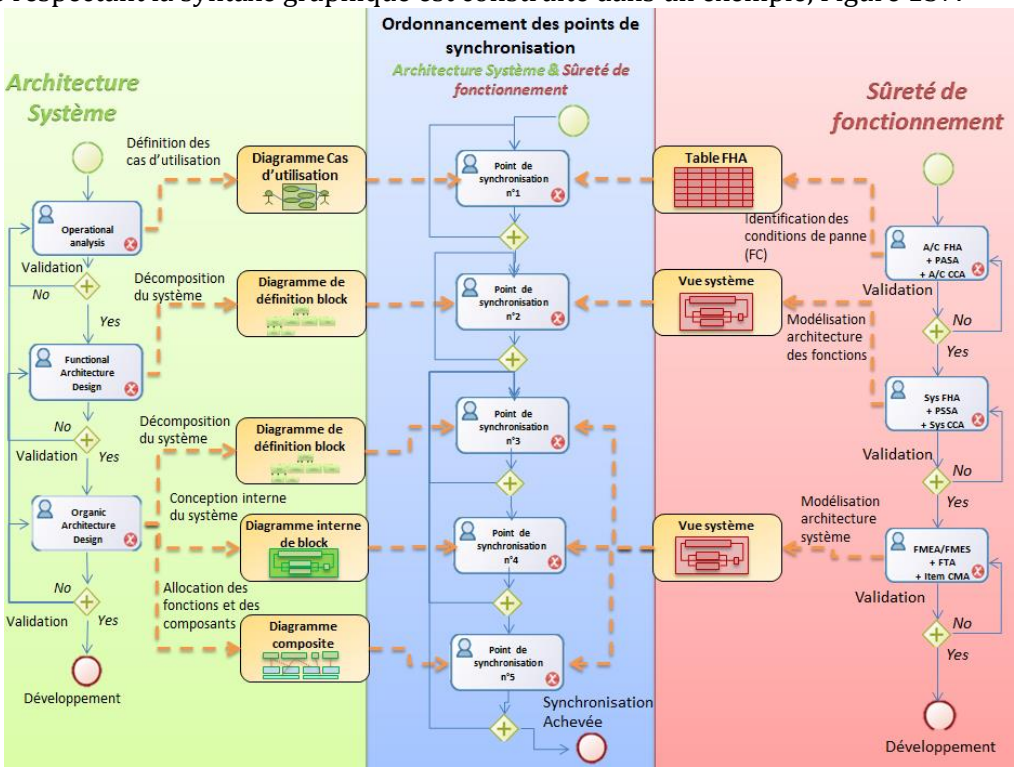


Figure 157 Exemple du vue d'ordonnancement des points de synchronisation

1.6. POINT DE VUE FONCTION DE TRANSFORMATION

Le point de vue concerne les « fonctions de transformation » utilisées pour caractériser l'exécution des transformations de modèles. Il fait référence au concept de « Transformation Map » du point de vue du « Mapping ». Chaque fonction définit, à partir de l'élément ou de la relation du métamodèle source considéré, le(s) élément(s) ou le(s) relation(s) du métamodèle cible correspondant, tout en capitalisant les données de traçabilité produites par la fonction.

$$F(X, Garde) = (X', Relation\ de\ traçabilité)$$
avec X un élément du métamodèle source et X' un élément du métamodèle cible

Le point de vue « fonction de transformation » est présenté Figure 158.

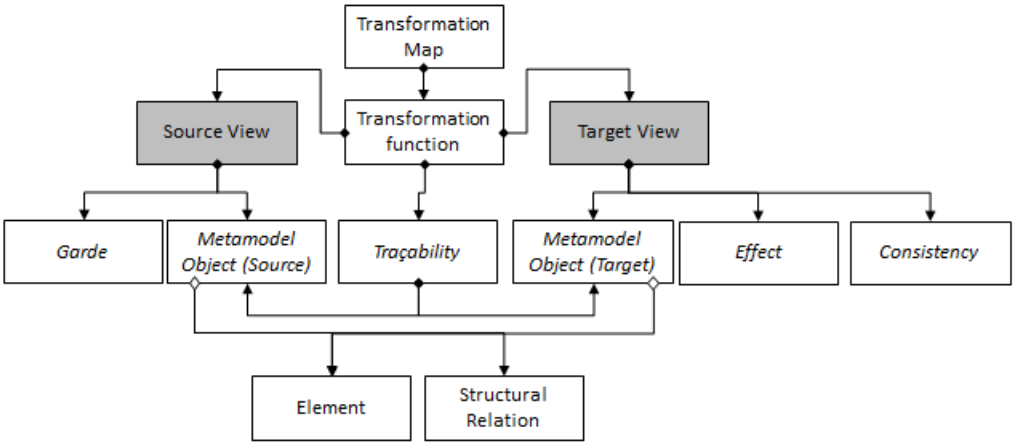
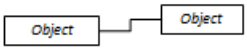
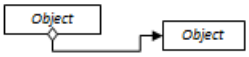


Figure 158 Métamodéle de « la fonction de tranformation » dans la synchronisation de modèles

A chaque élément du point de vue, une syntaxe graphique est proposée (élément graphique) aux éléments de la vue. Les équivalences sont représentées dans la Table 50. La syntaxe graphique s'inspire grandement des travaux d'Eugène Syriani sur l'outil ATomPM [120].

Table 50 Equivalence des concepts de fonction de transformation avec la syntaxe graphique

Concept du métamodèle correspondant	Syntaxe graphique des éléments de la vue	Concept du métamodèle correspondant	Syntaxe graphique des éléments de la vue
Transformation function	<Name>	Garde	« Garde description » Contenu dans une région vert.
Source View		Effect	« Effet » Contenu dans une région rouge.
Target View		Consistency	« Lien de cohérence » Contenu dans une région rouge.
Metamodel Object (Source)	 Contenu dans une région vert.	Traçability	 En interface des régions verte et rouge.
Metamodel Object (Target)	 Contenu dans une région rouge.		

Un template associé au point de vue, Figure 159, définit également les régions verte et rouge dans lesquelles certains concepts sont contraints.

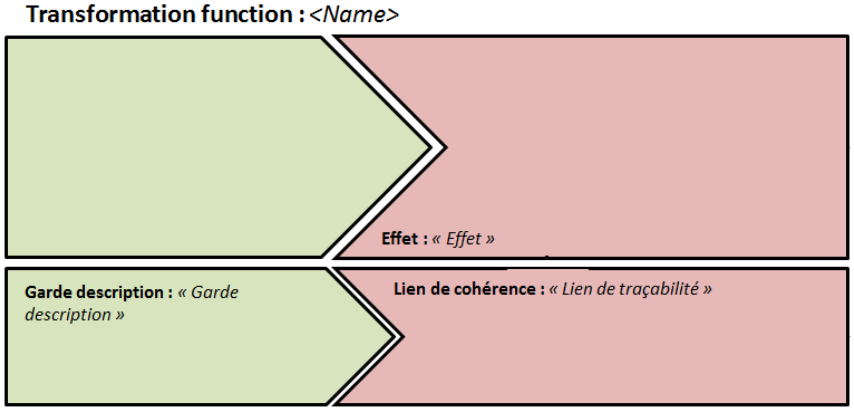


Figure 159 Template du point de vue "Fonction de transformation"

Les couleurs des régions sont indicatives, elles ne sont pas nécessairement représentées.

Un exemple de vue respectant le point de vue et l'application de la syntaxe graphique est présenté Figure 160.

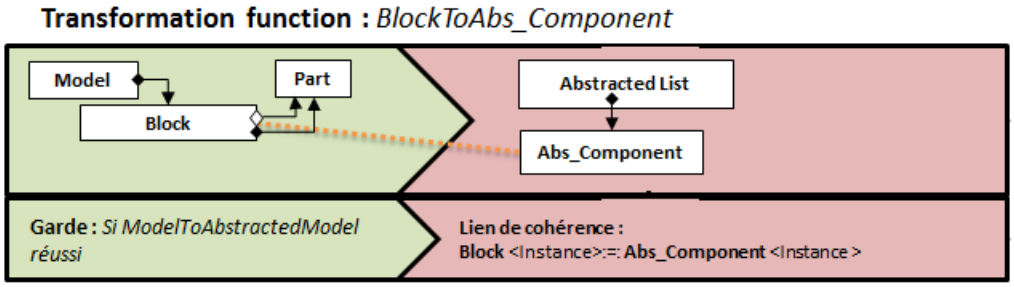


Figure 160 Exemple de vue d'une fonction de transformation

Deux exemples respectant le point de vue et la syntaxe associée sont utilisés dans le mémoire : Chapitre II6.3.2 et Chapitre II6.3.3.

2. ALGORITHMES DE COMPARAISON

Les comparaisons se déclinent en fonction du ou des concepts pivots d'architecture choisis. Elles sont essentiellement de deux types : la comparaison d'éléments et la comparaison de relations. Les deux algorithmes présentés ci-dessous décrivent le déroulement de chaque comparaison.

Les algorithmes présentés sont des exemples de « ce qu'il faudrait faire » pour comparer les contenus de modèles. Les travaux de thèse ne se sont pas intéressés aux techniques existantes. Des expérimentations additionnelles devraient être menées.

2.1. ALGORITHME DE COMPARAISON D'ENSEMBLES D'OBJETS

La comparaison d'éléments prend en entrée deux ensembles Ea et Eb. Les éléments sont abstraits et de même type dans les deux ensembles.

Données initiales :
On considère Ea l'ensemble des éléments du modèle abstrait du contexte A.

$$Ea = \{Ea_1, Ea_2, \dots, Ea_x\} \text{avec x} = \text{Cardinal}(Ea)$$

On considère Eb l'ensemble des éléments du modèle abstrait du contexte B.

$$Eb = \{Eb_1, Eb_2, \dots, Eb_y\} \text{avec y} = \text{Cardinal}(Eb)$$

L'algorithme, présenté Table 51, construit des relations de cohérence entre des éléments de même type (au même niveau d'abstraction).

Table 51 Algorithme de comparaison d'ensemble

1- Initialisation : i =1 et j=1,
2- Prendre l'élément Ea_i,
3- Interroger les utilisateurs sur les relations de cohérence entre Ea_1 *et* les éléments d'Eb
4- Lire et saisir la réponse de l'utilisateur (sous la forme d'une relation de cohérence),
5- Actualisation : i = i+1, Si (i <x) alors retourner à l'étape 2 sinon aller à l'étape 6.
6- Prendre l'élément Eb_j,
7- Interroger l'utilisation sur les relations de cohérence entre Eb_1 *et* les éléments d'Ea,
8- Lire et saisir la réponse de l'utilisateur (sous la forme d'une cohérence),
9- Actualisation : j=j+1, Si (j <y) alors retourner à l'étape 6 sinon aller à l'étape 10.
10- Récupérer toutes les relations de cohérence et construire un modèle de cohérence

Exemple simple : Une comparaison est appliquée sur deux ensembles de fonctions opérationnelles d'une voiture. Les données initiales sont les suivants:

- Ea et Eb sont deux ensembles de fonctions de modèles abstraits respectivement du contexte A et B.

$$Ea = \{Demarrer, Accelerer, Freiner, Changer\ de\ vitesse\}$$
$$Eb = \{Allumer\ le\ moteur, Se\ déplacer, Eteindre\ le\ moteur\}$$

L'exécution de l'algorithme est présenté Table 52.

Table 52 Exemple d'exécution de la comparaison des ensembles Ea et Eb

1	Initialisation	i =1 et j=1
2.1	Prendre Ea1	SujetComparaison = $Demarrer$
3.1	Interroger l'utilisation sur les relations de cohérence	L'utilisateur répond que le sujet est cohérent avec Eb1 « Allumer le moteur ». Il précise le sujet de la cohérence : « Démarrage voiture »
4.1	**Saisir la réponse**	$Co1 = \{Demarrage\ voiture, Re\}, avec\ Re = \{Re_1, Re_2\},$ $et\ Re_1 = \{Demarrer, 1\} et\ Re_2 = \{Allumer\ le\ moteur, unknow\}$
5.1	Actualisation	i=2, retour à l'étape 2
2.2	Prendre Ea2	SujetComparaison = $Accelerer$
3.2	Interroger l'utilisation sur les relations de cohérence	L'utilisateur répond que le sujet est cohérent avec Eb2 « Se déplacer ». Il précise le sujet de la cohérence : « En mouvement »
4.2	**Saisir la réponse**	$Co2 = \{En\ mouvement, Re\}, avec\ Re = \{Re_1, Re_2\},$ $et\ \ Re_1 = \{Accelerer, 1\} et\ Re_2 = \{Se\ déplacer, unknow\}$
5.2	Actualisation	i=3, retour à l'étape 2
2.3	Prendre Ea3	SujetComparaison = $Freiner$
3.3	Interroger l'utilisation sur les relations de cohérence	L'utilisateur répond que le sujet est cohérent avec Eb2 « Se déplacer ». Il précise le sujet de la cohérence : « En mouvement »
4.3	**Saisir la réponse**	$Co2 = \{En\ mouvement, Re\}, avec\ Re = \{Re_1, Re_2, Re_3\},$ $et\ Re_1 = \{Accelerer, 1\}, Re_2 = \{Freiner, 1\}\ et\ Re_3 = \{Se\ déplacer, unknow\}$
5.3	Actualisation	i=4, retour à l'étape 2
2.4	Prendre Ea4	SujetComparaison = $Changer\ de\ vitesse$
3.4	Interroger l'utilisation sur les relations de cohérence	L'utilisateur répond que le sujet n'as pas de cohérent avec Eb. Il précise le sujet de l'incohérence : « Changer de vitesse »
4.4	**Saisir la réponse**	$InCo1 = \{Changer\ de\ vitesse, Re\}, avec\ Re = \{Re_1, Re_2\},$ $Re_1 = \{Changer\ de\ vitesse, 1\}\ et\ Re_2 = \{\oslash\}$
5.4	Actualisation	i=5, aller à l'étape 6
6.1	Prendre Eb1	SujetComparaison = $Allumer\ le\ moteur$
7.1	Interroger l'utilisation sur les relations de cohérence	L'utilisateur répond que le sujet est cohérent avec Ea1 « Allumer». Il précise le sujet de la cohérence : « Démarrage voiture »
8.1	**Saisir la réponse**	$Co1 = \{Demarrage\ voiture, Re\}, avec\ Re = \{Re_1, Re_2\},$ $et\ Re_1 = \{Demarrer, 1\} et\ Re_2 = \{Allumer\ le\ moteur, 1\}$
9.1	Actualisation	j=2, retour à l'étape 6
6.2	Prendre Eb2	SujetComparaison = $Se\ déplacer$
7.2	Interroger l'utilisation sur les relations de cohérence	L'utilisateur répond que le sujet est cohérent avec Ea2 et Ea3 « Accelerer» et « Freiner ». Il précise le sujet de la cohérence : « En mouvement »
8.2	**Saisir la réponse**	$Co2 = \{En\ mouvement, Re\}, avec\ Re = \{Re_1, Re_2, Re_3\},$ $Re_1 = \{Accelerer, 1\}, Re_2 = \{Freiner, 1\}\ et\ Re_3 = \{Se\ déplacer, 1\}$
9.2	Actualisation	j=3, retour à l'étape 6
6.3	Prendre Eb2	SujetComparaison = $Eteindre\ le\ moteur$
7.3	Interroger l'utilisation sur les relations de cohérence	L'utilisateur répond que le sujet n'as pas de cohérent avec Ea. Il précise le sujet de l'incohérence : « Eteindre le moteur »
8.3	**Saisir la réponse**	$InCo2 = \{Eteindre\ le\ moteur, Re\}, avec\ Re = \{Re_1, Re_2\},$ $Re_1 = \{\oslash\}\ et\ Re_2 = \{Eteindre\ le\ moteur, 1\}$
9.3	Actualisation	j=4, retour à l'étape 6
10.1	Construire un modèle de cohérence	Modèle de cohérence = Concaténation des relations de cohérence et incohérences construites = $Co1 + Co2 + InCo1 + InCo2$

Pour considérer les objets relationnels, un second algorithme a été construit. Ces deux algorithmes associés correctement permettront de construire des comparaisons ciblées sur un concept pivot d'architecture.

2.2. ALGORITHME DE COMPARAISON DES RELATIONS

L'algorithme de comparaison de relations va permettre tout comme l'algorithme précédent de construire des relations de cohérence. La comparaison de relations se déroule de la même manière que pour celle des éléments. Cependant les données initiales sont structurées différemment car elles contiennent les objets relationnels et les éléments qu'elles lient.

$ObjeE$ = objectif d'étude, $ObjeE$ = paramètre du Cpa

R_{struc} = Relations de structuration considérés, $R_{struc} = \{R_{struc\,1}, \dots, R_{struc\,n}\}$, avec $n \in \mathbb{N}^*$

E_{type} = Type des éléments considérés, $E_{type} = \{E_{type\,1}, \dots, E_{type\,m}\}$, avec $m \in \mathbb{N}^*$

$Cpa = concept\ pivot\ d'architecture$, $Cpa_{ObjetE} \subseteq R_{struc} \times E_{type}$

Données initiales :

- On considère Ma un modèle contenant un ensemble d'éléments Ea et un ensemble de relations Ra.

  Ea = un ensemble d'éléments $= \{Ea_1, Ea_2, \dots, Ea_x\}$, avec x = Cardinal(Ea)

  $et\ Ra$ = un ensemble de relations $= \{Ra_1, Ra_2, \dots, Ra_y\}, avec\ y = Cardinal(Ra)$

  $$Ma = Ea \times Ra$$

- On considère Mb un modèle contenant un ensemble d'éléments Eb et un ensemble de relations Ra.

  Eb = un ensemble d'éléments $= \{Eb_1, Eb_2, \dots, Eb_x\}$, avec x = Cardinal(Eb)

  Eb = un ensemble de relations $= \{Rb_1, Rb_2, \dots, Rb_y\}, avec\ y = Cardinal(Ra)$

  $$Ma = Ea \times Ra$$

Note : Un objet relationnel Ra_h est défini par un élément de Ea source et Ea cible.

L'algorithme, Table 53, décrit la mise en cohérence entre des relations de même type. En spécialisant les relations, il est possible de définir plusieurs types de comparaisons relationnelles, notamment une comparaison de relation d'agrégation, de composition, de connexion ou d'héritage.

Table 53 Algorithme de comparaison des relations

1- Initialisation : i =1 et j=1,
2- Prendre l'élément Ra_i,
3- Interroger les utilisateurs sur les relations de cohérence entre Ra_1 *et* les éléments de Rb,
4- Lire et saisir la réponse de l'utilisateur (sous la forme d'une relation de cohérence),
5- Actualisation : i = i+1, Si (i <x) alors retourner à l'étape 2 sinon aller à l'étape 6.
6- Prendre l'élément Rb_j,
7- Interroger l'utilisation sur les relations de cohérence entre Rb_1 *et* les éléments d'Ra,
8- Lire et saisir la réponse de l'utilisateur (sous la forme d'une relation de cohérence),
9- Actualisation : j=j+1, Si (j <y) alors retourner à l'étape 6 sinon aller à l'étape 10.
10- Récupérer toutes les relations de cohérence et construire un modèle de cohérence

3. TRANSFORMATION SYSML VERS ALTARICA 3.0

L'outil Sophia [74], [73] développé par le CEA au laboratoire LISE est un outil basé sur la plateforme Papyrus. Il utilise des profils pour construire des points de vue d'analyse 'safety' en relation avec différents standards (selon les étapes des processus plusieurs types d'analyses peuvent être conduites : PHA, FTA, FMEA, …). Ces vues orientées safety permettent de saisir des données et des modèles d'analyse dysfonctionnelle sur la base d'une architecture issue

directement du modèle de conception par abstraction simple. Le principe de modélisation de l'architecture s'appuie sur les concepts de structuration d'UML/SysML. Toutefois les langages d'entrée peuvent être des langages spécifiques métier (RobotML pour la robotique [11] par exemple). De même le principe d'annotation peut s'appliquer sur d'autres langages (comme EAST-ADL). L'ingénieur 'safety' construit donc sur cette base une vue dysfonctionnelle. L'exploitation de ces informations est ensuite faite à l'aide d'outils formels. Des transformations dédiées ont été réalisées vers les formalismes AltaRica Dataflow et NuSMV. Pour calculer des coupes minimales par exemple, on traduit le modèle en AltaRica Dataflow et on utilise des modules tels que l'outil ARC du LaBRI ou xFTA [137] de l'association AltaRica.

La transformation vers AltaRica 3.0 [70], [71] a été réalisée dans la cadre de cette thèse. Le programme initialement réalisé pour le langage AltaRica Dataflow a été repris et enrichi pour le langage AltaRica 3.0.

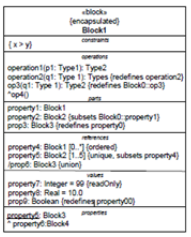
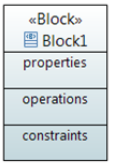
Le programme est présenté en Annexe 3.1. Il est écrit en Java et contient 9 classes :

- **TransformationModelToAltaRica** : récupère le modèle source et l'élément sommet. Elle définit ensuite un objet de la classes modelAltaRica à partir de la classe ModelToAltaRicaModel.java puis appelle la génération du modèle AltaRica 3.0 et restitue le code généré ;
- **ModelToAltaRicaModel :** réalise la génération du modèle AltaRica 3.0. Elle recherche l'élément sommet du modèle à partir de la classe Alt_NodeMain, puis construit la hiérarchie des composants au fur et à mesure qu'ils sont générés et retourne le modèle fourni par la classe SystemAltaRica ;
- **Alt_NodeMain :** détermine l'élément sommet du modèle en AltaRica 3.0. Dans l'ordre, elle définit les sous-éléments, les variables de flux (input/output) puis ajoute si nécessaire des éléments fictifs en entrée de l'élément sommet et les assertions ;
- **Alt_Atomic_Node :** définit chaque élément de plus bas niveau en classe AltaRica 3.0, notamment la déclaration de la classe, les variables internes, les variables de flux et des évènements. Elle définit ensuite les transitions et les assertions. Elle retourne à SystemAltaRica, la classe au format AltaRica 3.0 ;
- **IntermediateNode :** définit chaque élément de niveau intermédiaire en une classe AltaRica 3.0. Elle déclare la classe et des variables de flux puis poursuit par les assertions. Elle retourne l'expression de la classe au format AltaRica 3.0. *La transformation de modèles fait l'hypothèse que les éléments intermédiaires n'ont pas de comportement propre.*
- **Alt_Port :** retourne les propriétés des éléments de type Port (SysML), comme le « name », « direction » ou « id ») ;
- **Alt_StateVariable :** retourne le nom des éléments de type State (SysML) ;
- **SystemAltaRica :** définit l'assemblage des parties du modèle en AltaRica 3.0. Elle commence par assembler les classes fictives, puis l'ensemble des éléments de bas niveau, puis l'ensemble des éléments de niveau(x) intermédiaire(s) et enfin la classe sommet.
  Les classes fictives ne sont pas utiles pour la transformation ni pour les calculs dédiés. Elles ont été conservées des travaux précédents, d'autres solutions beaucoup moins couteuses en code existent.

Le programme a été effectué au début de la thèse. Il reste avant tout un prototype pour montrer les capacités de transformation de SysML vers AltaRica 3.0. Il contient de nombreuses limites, e.g. il ne permet que de déclarer des variables booléennes pour caractériser l'état du composant, il ne considère que des assertions orientées, etc.

Il a permis de comprendre les limites des transformations de modèles. Les différences de nature, de syntaxe et de sémantique des langages ont amené de nombreux questionnements. Il a notamment permis de comprendre que la transformation de modèle peut seulement retranscrire l'intersection (en opposition à l'union) des sémantiques des langages.

Les concepts du langage SysML et AltaRica 3.0 ont été cartographiés. La Figure 161, illustrée par un exemple, représente les associations entre les objets des langages.

Objet AltaRica 3.0	Syntaxe AltaRica 3.0	Exemple AltaRica 3.0	Objet SysML (ou UML)	Syntaxe SysML	Exemple SysML sur Papyrus
<Block>	block <Block> <Body> Ou <Class> <Class> Ou <Block> <Block> end	block Voiture ... end	SysML::Blocks::Block Ou SysML::Blocks::Part		
<Class>	class <Class> <Body> Ou <Class> <Class> end	class NonRepairableComponent ... end	SysML::Blocks::Block		

<Body>	<Vf>
	<BehaviorPart>
	<Assertion>
		Boolean inflow (reset = false);
		Boolean outflow (reset = false);
		Boolean working (init = true);
		event failure;
		transition
		failure : working -> working :=false;
		assertion
		outflow := inflow and working;
			SysML::Blocks::Block
			PROFIL SOPHIA
			Stereotype Top:Boolean		

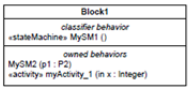
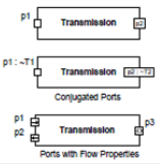
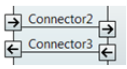
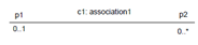
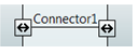
<BehaviorPart>	<Vi><Event><Transition>	Boolean inflow (reset = false); Boolean outflow (reset = false); event failure; transition failure : working -> working :=false;	PROFIL SOPHIA Stereotype FTAQualitative		
<Var> de type <Vi> Variable interne	<Variable> <Vi> (init = <Variable>);	Boolean working (init = true);	SysML::Blocks::Block PROFIL SOPHIA Stereotype ::failureMode Fm_i Automatique build (tolerence du prototypage)	workingFm_i :Boolean	working
<Var> de type <Vf> Variable de flux	<Variable> <Vf> (reset = <Variable>);	Boolean inflow (reset = false);	UML4SysML::Port		
<Event>	event <Event>;	event failure;	SysML::Blocks::::Block PROFIL SOPHIA Stereotype ::failureMode Fm_i Automatique build (tolerence du prototypage)	failureFm_i	failureFm_1
<Transition> ::	transition <Event>: <Condition> -> <Effet>;	transition failure : working -> working :=false;	SysML::Blocks::::Block PROFIL SOPHIA Stereotype ::failureMode Fm_i Automatique build (tolerence du prototypage)	transition failureFm_i : working Fm_i-> workingFm_i :=false;	transition failureFm_1 : working Fm_1-> workingFm_1 :=false;
<Condition>	F(<Var>)= boolean expression (Not, if then else, (!, ?),&, |, and or).	working if state1 then ...	SysML::Blocks::::Block Stereotype ::failureMode Fm_i Automatique build (tolerence du prototypage)	working Fm_i	working Fm_1
<Effet>	<Var> == <Var>	working := false			
<Assertion> :: type Behavioral	"<Condition><Vf> := <Condition><Effet>;"	"out := in and working;" Ou "if state1 then Block1.Out := 1 else Block1.Out := 0;"	UML4SysML::Port PROFIL SOPHIA Stereotype:: deviation expression Automatique build (tolerence du prototypage)	workingFm_i :=false	workingFm_1 :=false
Assertion :: type Block	<Block>.<vf> := <Block>.<vf>	Block1.In := Block2.Out;	UML4SysML::Connector UnidirectionalConnector UML4SysML::Port PROFIL SOPHIA Stereotype:: deviation Expression		
Assertion :: type Block	<Block>.<vf> :=: <Block>.<vf>	Block1.In :=: Block2.Out;	UML4SysML::Connector BidirectionalConnector UML4SysML::Port PROFIL SOPHIA Stereotype:: deviation Expression		

Figure 161 Equivalence des concepts SysML et AltaRica 3.0 pour une vue d'architecture système physique cohérente

4. ABSTRACTION QVT-O

Les transformations présentées peuvent être réutilisées et mise dans une bibliothèque des Mappings. Quatre abstractions ont été développées, elles sont détaillées dans cette partie.

4.1. TRANSFORMATION DE MODELES USECASE VERS UNE LISTE ABSTRAITE

Les diagrammes de cas d'utilisation sont utilisés pour représenter des vues opérationnelles du système, e.g. Chapitre V point de synchronisation n°1. Une transformation de modèles QVT-O [117] a été développée. Elle lit un modèle UML et construit un modèle listant les cas d'utilisation.

La transformation tient compte des métamodèles source, UML 5.0 [26] et cible, modèle d'objet sémantique (cf. Figure 124). Le mapping entre ces deux métamodèles est construit dans la Figure 162.

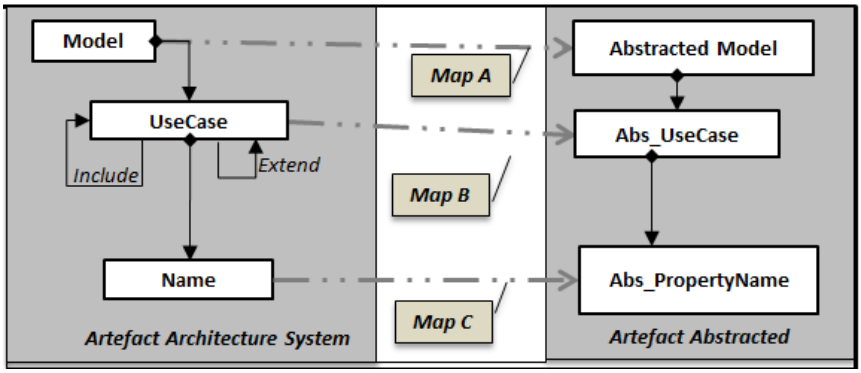


Figure 162 Mapping de modèles de cas d'utilisation vers une liste abstraite

Les trois relations *Map* précédentes sont définies par les fonctions de transformation suivantes : ***ModelToAbstractModel, UseCaseToAbs_UseCase, NameToAbs_PropertyName***. Elles sont décrites, Figure 163, par le point de vue «Fonction de Transformation » et leur ordonnancement (par un diagramme BPMN).

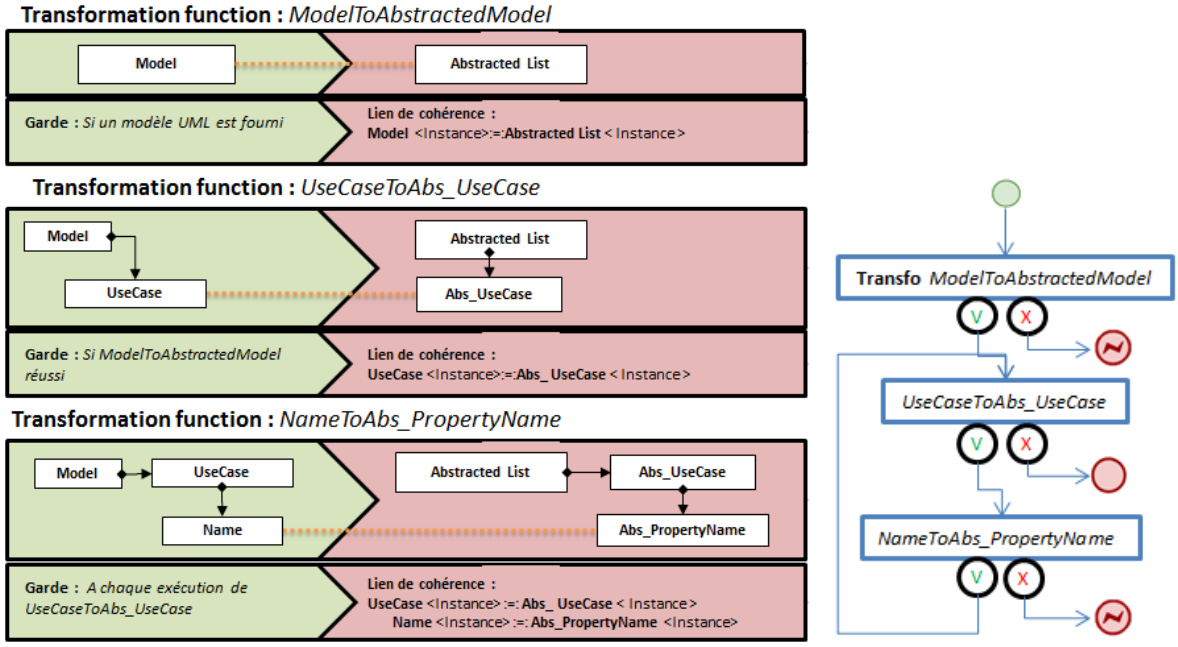


Figure 163 Fonctions de transformation de modèles et leur ordonnancement

Le programme est présenté en Annexe 3.2, il est exprimé en QVT-O [117]. Il prend en entrée deux métamodèles (source UML [26] et cible AbstractedList) et exécute la class main.

4.2. TRANSFORMATION DE MODELES BDD VERS DES MODELES HIERARCHISES

Le diagramme 'Block Définition' est un point de vue très employé pour décrire la décomposition d'un système. I.e. Chapitre V point de synchronisation n°3. Une transformation de modèles basée sur le langage QVT-O [117] a été développée. Cette transformation va lire un modèle UML fourni par l'utilisateur et construire un modèle listant les cas d'utilisation.

La transformation tient compte des métamodèles source, UML 5.0 et cible, modèles de relation de composition sur des composants (cf. Figure 130). Le mapping des deux métamodèles est présenté à la Figure 164.

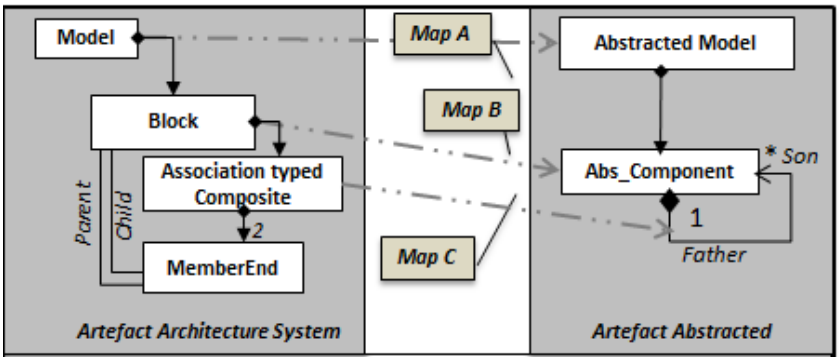


Figure 164 Mapping de modèles BDD vers un modèle hiérarchisé et abstrait

Les trois relations *Map* précédentes sont définies par les fonctions de transformation suivantes : ***ModelToAbstractModel, BlockToAbs_Component, CompositionToAbs_Composition***. Elles sont décrites, Figure 165, par le point de vue «Fonction de Transformation » et leur ordonnancement (par un diagramme BPMN).

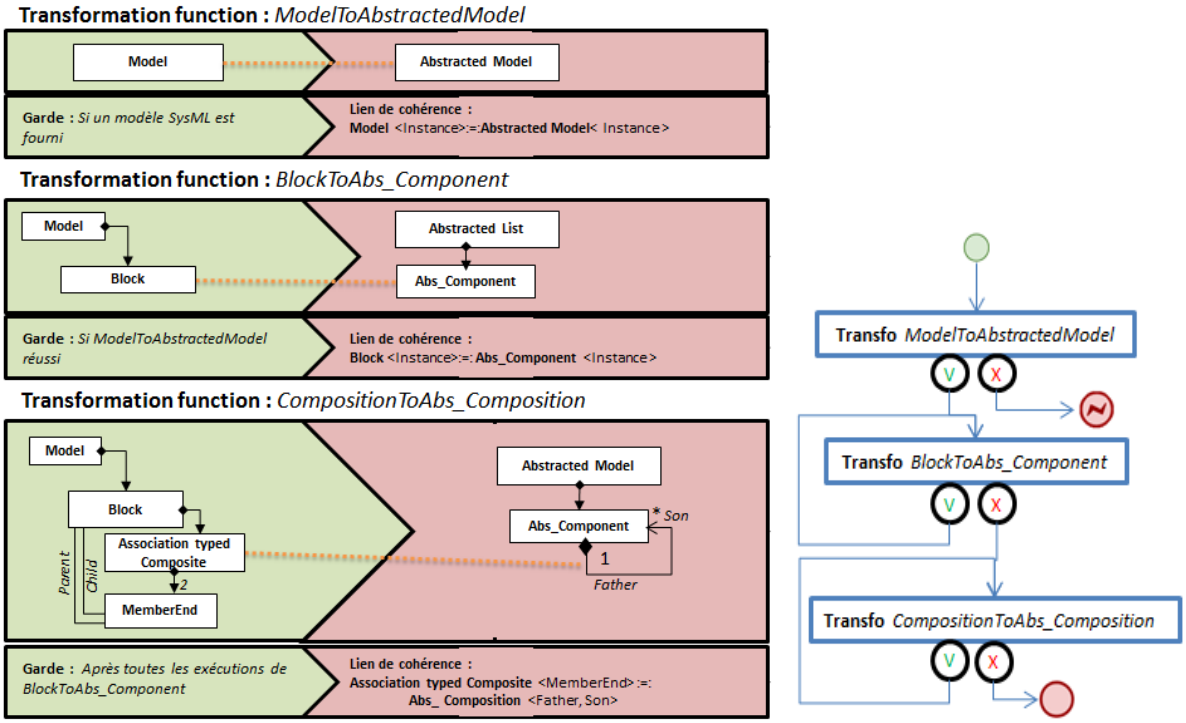


Figure 165 Fonctions de transformation de modèles et leur ordonnancement

Le programme est présenté en Annexe 3.3, il est exprimé en QVT-O, un langage de transformation de modèles. Il prend en entrée deux métamodèles (source UML et cible AbstractedList) et exécute la class main.

4.3. TRANSFORMATION DE MODELES IBD VERS DES MODELES DE CONNEXIONS

Le diagramme Interne Block Diagramme est un point de vue très employé pour décrire la connectique des composantes d'un système. I.e. Chapitre V point de synchronisation n°4. Le concept de block et de relation de connexions sont couramment utilisés. Une transformation de modèles basée sur langage QVT-O a été développée. La transformation va lire un modèle UML fourni par l'utilisateur et construire un modèle de la hiérarchie des composants du système.

La transformation tient compte des métamodèles source, UML 5.0 et cible, modèle de relation de composition sur des composants (cf. Figure 133). Le mapping des deux métamodèles est présenté Figure 166.

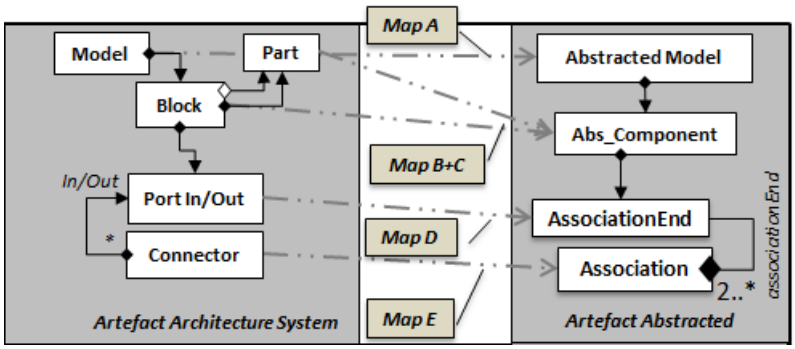


Figure 166 Mapping de modèles IBD vers un modèle d'association et abstrait

Les quatre relations *Map* précédentes sont définies par les fonctions de transformation suivantes :

- ***ModelToAbstractModel,***
- ***BlockToAbs_Component,***
- ***MemberEndToAbs_AssociationEnd***.

Elles sont décrites, Figure 167, par le point de vue «Fonction de Transformation » et leur ordonnancement (par un diagramme BPMN).

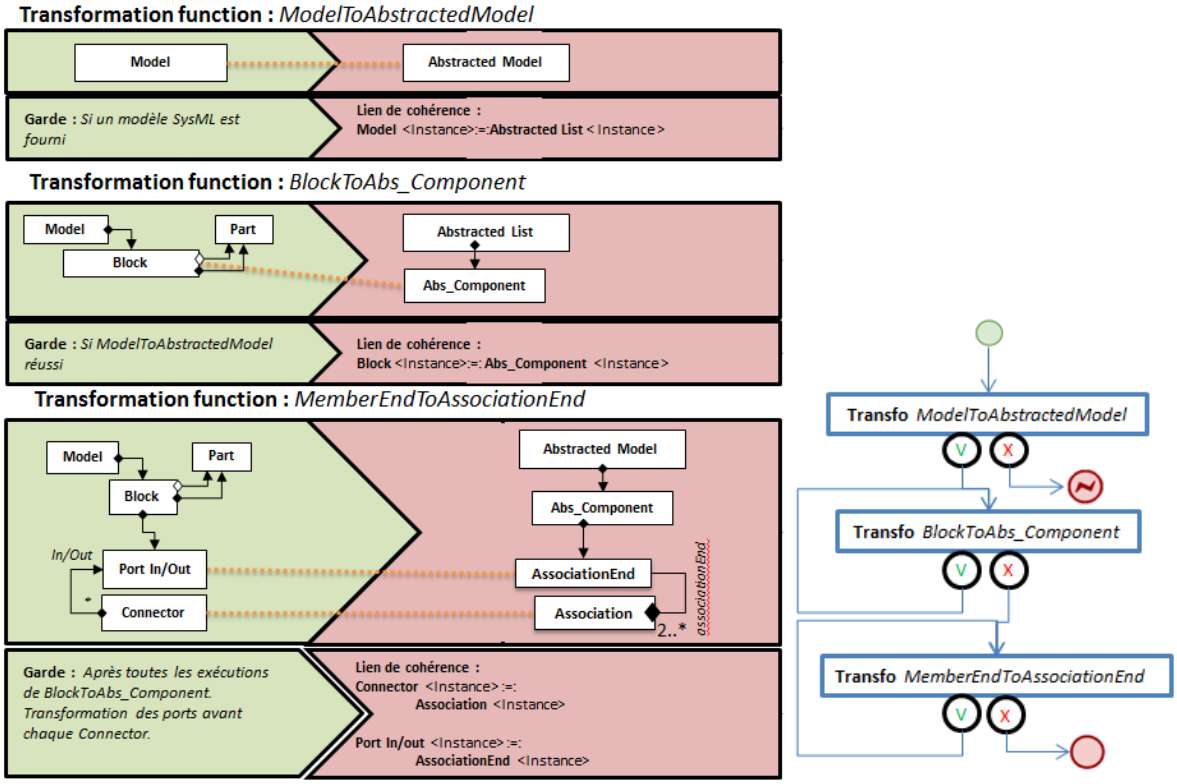


Figure 167 Fonctions de transformation de modèles et leur ordonnancement

Le programme est présenté en Annexe 3.4, il est exprimé en QVT-O un langage de transformation de modèles. Il prend en entrée deux métamodèles (source UML et cible AbstractedList) et exécute la class main.

Remarque : l'abstraction ne porte ici que sur des modèles UML ou SysML. D'autres formalismes peuvent être considérés, il faut cependant pouvoir l'exprimer au format Ecore ou UML.

4.4. TRANSFORMATION DE MODELES ALTARICA 3.0 VERS DES MODELES DE COMPOSITION

Les modèles spécifiés en AltaRica 3.0 [70] ne permettent pas d'utiliser directement QVT-O pour réaliser des abstractions par transformation de modèles. Il faudrait que le langage AltaRica 3.0 soit caractérisé par le MOF [139].

Un prototype en Python a été réalisé (cf. Programme 1) pour illustrer un cas d'abstraction. Il va lire un modèle AltaRica 3.0 et ne retenir que les noms des composants instanciés. L'exemple ci-dessous présente un modèle simple en AltaRica 3.0 en entrée de la transformation.

```
class RepairableComponent
    Boolean working (init = true);
    parameter Real lambda = 0.0001;
    parameter Real mu = 0.01;
    event failure (delay = exponential(lambda));
    event repair (delay = exponential(mu));
    transition
        failure: working -> working := false;
        repair: not working -> working := true;
end

class Component
    extends RepairableComponent;
    Boolean vfInput, vfOutput (reset = false);
    assertion
        vfOutput := if working then vfInput else false;
end

block Artefact
    Component a,b,c,d,e;
    observer Boolean oOut = e.vfOutput;
    assertion
        a.vfInput := true;
        b.vfInput := a.vfOutput;
        c.vfInput := b.vfOutput;
        d.vfInput := a.vfOutput;
        e.vfInput := c.vfOutput or d.vfOutput;
end
```

Programme 1 Extrait de modèle AltaRica 3.0 utilisé pour le programme de transformation Python

La transformation réalisée en python est construite en 5 parties. Une première importe les bibliothèques nécessaires à l'exécution de fonctions génériques. La seconde partie définit la classe *ComponentList*. Elle permet de caractériser les composants et leur type. La troisième partie permet de construire un modèle XMI à partir des données de la classe *ComponentList.* La quatrième partie définit la classe *AltaRicaReader* qui lit le fichier *source* au format AltaRica 3.0. Enfin, la dernière partie du programme intègre les classes entre elles. Le programme est présenté en Annexe 3.5.

Remarques, le prototypage présenté n'est pas une solution optimisée et peut être enrichi. Les API python tels que SAX ou DOM permettent de manipuler les objets des modèles de façon plus rapide et intuitive.

L'exécution de ce programme génère un document *target.xmi* pouvant être utilisé pour la comparaison de modèles. Le résultat produit (*target.xmi*) est contenu dans le Programme 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<mmsource:root xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mmsource="http://www.mmsource.fr" xsi:schemaLocation="http://www.mmsource.fr
../MModelSource/MMSource.ecore">
  <element xsi:type="mmsource:Component" classblock ="Component"/>
  <element xsi:type="mmsource:Component" classblock ="a"/>
  <element xsi:type="mmsource:Component" classblock ="b"/>
  <element xsi:type="mmsource:Component" classblock ="c"/>
  <element xsi:type="mmsource:Component" classblock ="d"/>
  <element xsi:type="mmsource:Component" classblock ="e"/>
</mmsource:root>
```

Programme 2 Résultat de transformation de modèles AltaRica 3.0 vers des listes de composants en Ecore

5. CONCRETISATION

La concrétisation effectuée est très simple. Elle permet d'ajouter un élément de type commentaire à un élément du modèle. Le contenu de ce commentaire est une description informelle de compromis. Il est envisageable qu'il contienne des objets plus sophistiqués comme un pattern.

La transformation de modèles lie un modèle source de type UML et un « modèle de commentaire » (comme liste de compromis). Elle reconstruit le modèle UML en ajoutant les commentaires annotés sur les éléments correspondants.

La Figure 168 illustre un exemple de modèle UML source, il est fourni par l'ingénieur avant l'abstraction. Il est de nouveau employé pour la concrétisation afin de rajouter des commentaires annotés sur des éléments ou des relations.

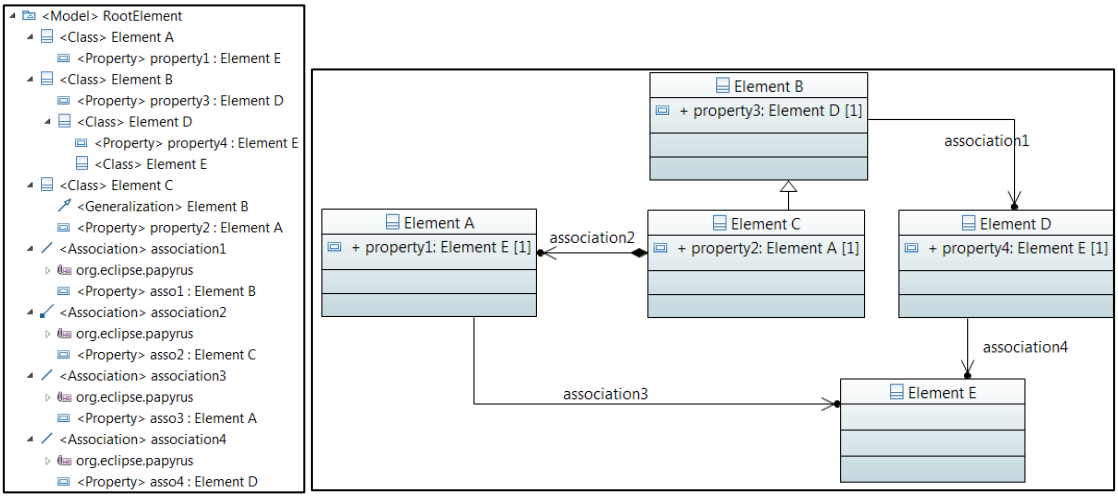


Figure 168 Modèle UML produit par une discipline

Le « modèle de commentaire », Figure 169, contient un ensemble de classes possédant une propriété référençant l'élément ciblé et un commentaire contenant la description du compromis. Celui-ci est proposé par les ingénieurs pour résoudre les incohérences identifiées.

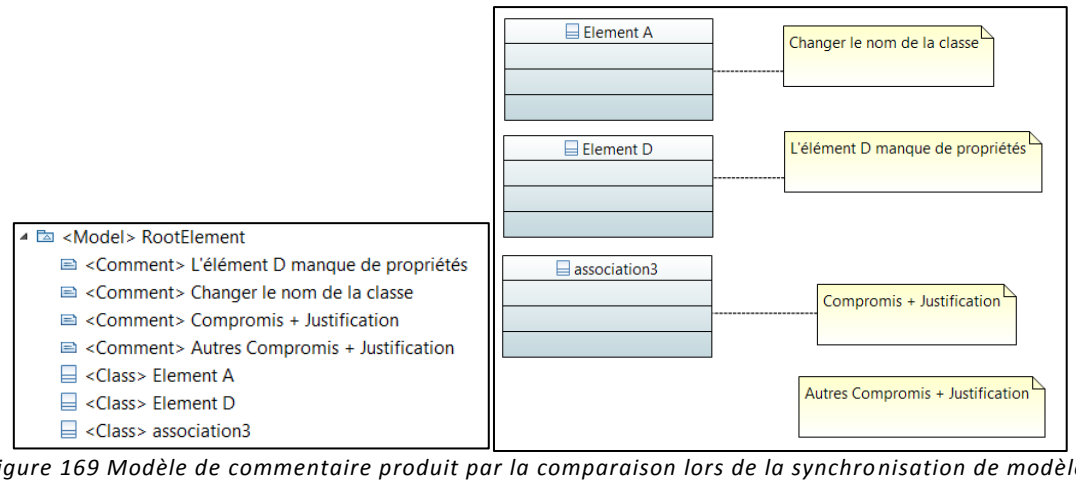


Figure 169 Modèle de commentaire produit par la comparaison lors de la synchronisation de modèles

L'objectif de la concrétisation est de convertir les deux modèles précédents en un seul modèle UML. Le programme est présenté en Annexe 3.6. Il applique 4 fonctions de transformation (mapping) : Model2Model, Class2Class, Asso2Asso, Gen2Gen. Ces fonctions font appel à des requêtes (Query) pour faciliter l'exploration des modèles.

L'exécution du programme QVT-O, reproduit le modèle en ajoutant des éléments commentaires en fonction du modèle de commentaire. Le résultat issu de l'exécution de ce programme est présenté Figure 170.

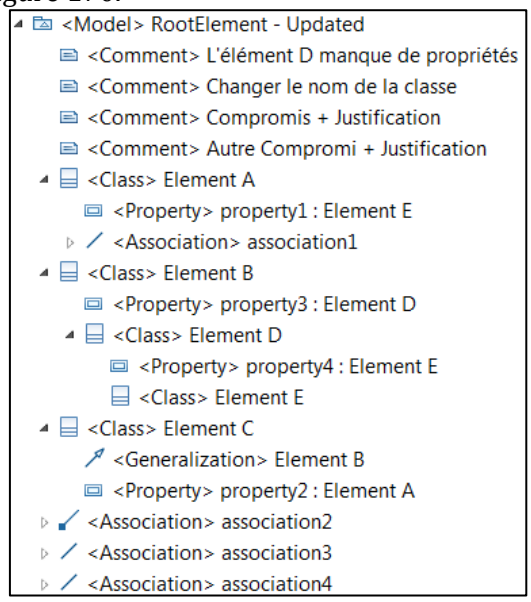


Figure 170 Modèle résultant de la concrétisation

La Figure 171 présente les propriétés des commentaires annotées sur l'élément A et D.

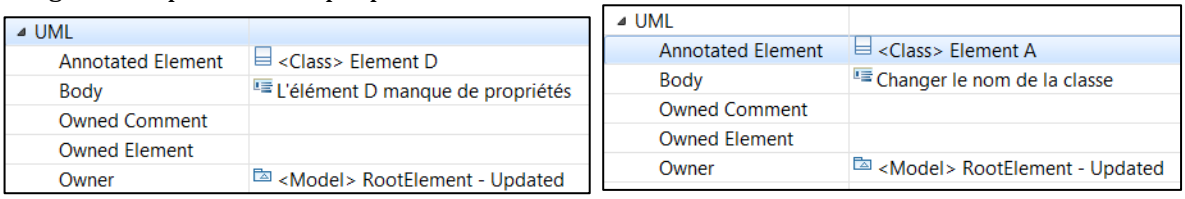


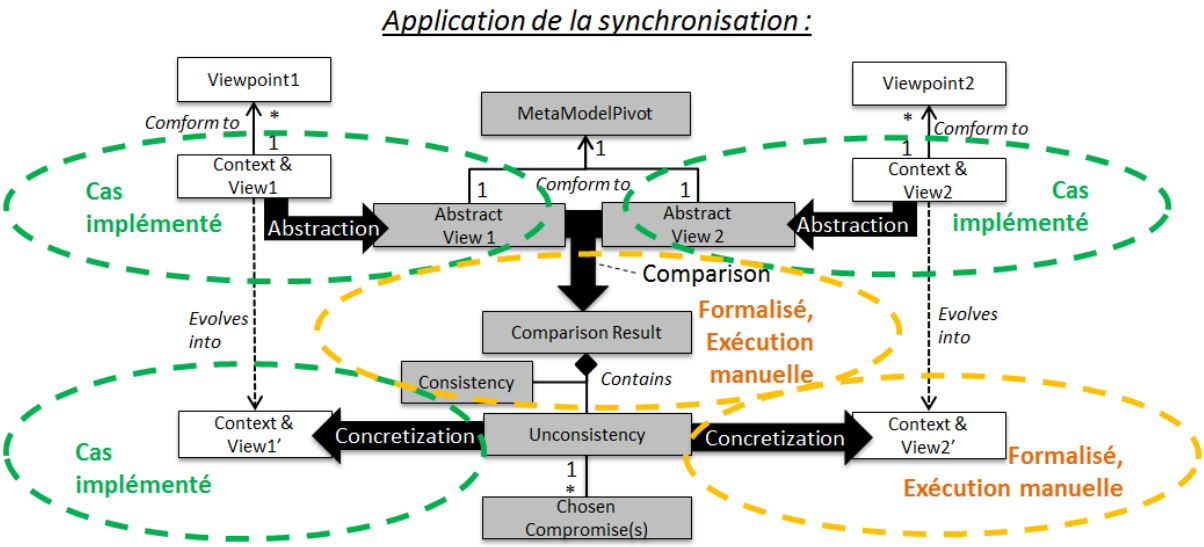
Figure 171 Propriétés des commentaires ajoutées au modèle source.

6. SYNTHESE DES TRAVAUX D'IMPLEMENTATION

Pour terminer ce chapitre, nous dressons un court bilan des développements réalisés. Les travaux d'implémentation ont permis d'obtenir les résultats suivants :

- 5 points de vue formalisés par une sémantique et une syntaxe graphique ;
- 1 profil implémenté sur Papyrus [134] pour définir les contextes des disciplines d'ingénierie ;
- 1 transformation de modèles en Java de « SysML+ Profil Sophia » vers « AltaRica 3.0 » ;
- 2 algorithmes de comparaison de modèles ;
- 1 mapping complet entre SysML et AltaRica 3.0 ;
- 3 implémentations d'abstraction à partir de diagramme UML ou SysML ;
- 1 implémentation d'abstraction à partir d'un modèle AltaRica 3.0 ;
- 1 implémentation de concrétisation vers un modèle UML.

Ces travaux ont démontré la faisabilité technologique d'un environnement outillé de synchronisation de modèles. La Figure 172 dresse le bilan des travaux d'implémentation des étapes fondamentales de la synchronisation (abstraction, comparaison, concrétisation) et des étapes de la méthodologie (définition des contextes d'ingénieries, configuration de la synchronisation, application de la synchronisation, etc.)



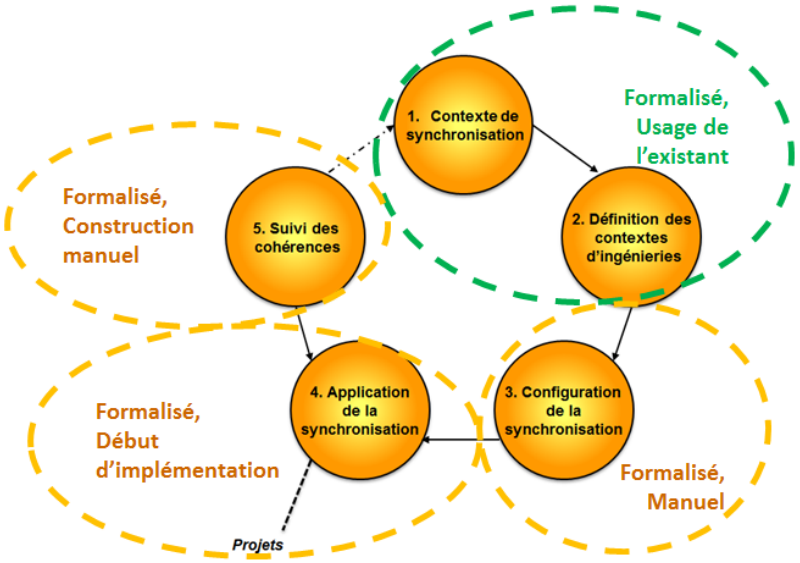


Figure 172 Bilan des formalisations et des implémentations

CONCLUSION

En introduction du manuscrit, nous avions identifié la problématique selon laquelle la synchronisation de modèles permettrait de solutionner les problèmes d'interactions multidisciplinaires, tout au long de son cycle de développement du système. Cet enjeu devient particulièrement important lorsqu'il traite de systèmes complexes. L'un des principaux objectifs était de définir une mise en œuvre de la synchronisation de modèles entre la conception d'architecture système et l'analyse de la sûreté de fonctionnement.

Tout au long du manuscrit, nous avons montré ce que la notion de synchronisation de modèles sous-tendait à différents niveaux de réflexion (au niveau conceptuel puis méthodologique et au niveau des applications). Le cadre conceptuel et la méthodologie proposés s'appuient sur une combinaison de solutions modulaires répondant à la problématique. L'application peut ainsi être construite de manière personnalisée selon les besoins et le contexte industriel.

L'intérêt du travail effectué était quadruple, ces contributions sont détaillées par la suite :

- L'apport d'un cadre conceptuel ;
- La proposition d'une méthodologie ;
- L'expérimentation sur le cas d'étude ;
- Les travaux d'implémentation.

En complément de la conclusion, une courte évaluation des contributions et des actions additionnelles effectuées durant ma thèse sont présentées en Annexe 4.

1. LE CADRE CONCEPTUEL

Le cadre conceptuel apporte l'ensemble des concepts et des paradigmes permettant la mise en œuvre, l'application et le suivi de la synchronisation de modèles. Au total, 44 concepts sont présentés. Ils sont tous rattachés à une définition et utilisés par un métamodèle ou un formalisme.

Les apports majeurs du cadre conceptuel sont les suivants :

- Le cadre a formalisé l'ensemble des fonctionnalités attendues d'un outil de synchronisation de modèles.
- Le cadre permet de mettre en œuvre les mécanismes de synchronisation de modèles initialement prévus : l'abstraction, la comparaison et la concrétisation.
- Il apporte cinq notions essentielles : le contexte d'une discipline d'ingénierie, le point de synchronisation, le concept pivot d'architecture, la mise en cohérence ainsi que la traçabilité.
- Le cadre a été utilisé pour définir la méthodologie de synchronisation de modèles.
- Les concepts des processus de synchronisation de modèles et des disciplines d'ingénierie ont été définis séparément. L'objectif est d'encourager la réutilisation des concepts dans de futurs travaux de recherche liés à la mise en cohérence de modèles hétérogènes ou aux interactions multidisciplinaires.

Le cadre conceptuel ouvre les perspectives suivantes :

Perspective 1 *Un renforcement des concepts d'architecture pivot, du concept de métamodèles pivots et de leurs dépendances permettrait d'identifier un ou plusieurs langage(s) pivot(s) pour la synchronisation de modèles qui pourrait(ent) s'adapter au niveau d'abstraction choisi.*

Perspective 2 *Les concepts de la traçabilité pourraient être d'avantage formalisés afin de spécifier formellement leurs constructions, leurs interprétations et leurs dépendances avec les modèles de cohérence.*

2. LA METHODOLOGIE

La méthodologie proposée est itérative. Elle contient 5 étapes permettant de définir, de caractériser, de configurer, d'appliquer et de suivre la synchronisation de modèles dans un projet de conception de système.

Les apports majeurs de la méthodologie sont :

- La formalisation des étapes va permettre l'application d'interactions multidisciplinaires en entreprise.
- Ces étapes ont été définies indépendamment des préoccupations génie logiciel telles que les choix d'implémentation des langages, des environnements de modélisation et des techniques de développement logiciel.
- Bien que les problèmes d'interaction entre la conception d'architecture système et la sûreté de fonctionnement aient motivé ce travail, la méthodologie proposée est indépendante des disciplines étudiées. Elle pourrait donc s'utiliser pour d'autres interactions multidisciplinaires s'intéressant à l'architecture du système.
- La méthodologie a été appliquée sur un cas d'étude industriel concernant un système de détection incendie pour faire ressortir ses caractéristiques et ses limites.
- La méthodologie est conçue pour s'adapter au contexte de l'entreprise par l'ajout ou la modification d'étape et méthode.

La méthodologie présente cependant quelques limites :

- Elle est couteuse en ressources durant le premier cycle itératif mais apporte un retour d'investissement significatif par la suite. Ceci est lié au fait qu'il est nécessaire de capitaliser une grande partie de la connaissance des processus, des méthodes, des modèles, etc., employés par les disciplines d'ingénierie concernées.
- Son application n'est pas pertinente pour des projets à court terme ou pour des petites entreprises n'ayant pas de disciplines d'ingénierie distinctes.
- La mise en pratique d'une telle méthodologie (même outillée) ne peut être envisagée sans que l'entreprise mène une importante conduite du changement (au niveau humain ainsi qu'au niveau de ses démarches model-based).

La méthodologie ouvre les perspectives suivantes :

Perspective 3 *La conduite du changement reste un défi majeur dans le cadre de ces travaux. Une formalisation et une organisation des évolutions associées à la méthodologie pourraient permettre la réalisation des vues, des méthodes et des points de synchronisation vers une démarche model-based de plus en plus mature.*

Perspective 4 *Cette démarche novatrice est une rupture avec les démarches actuelles* [20]*. Elle impose aux disciplines d'ingénierie une collaboration forte et conjointe tout en conservant la séparation des préoccupations durant les études. Des recherches plus sociales ou sociétales sur les évolutions nécessaires de la culture d'entreprise, pourraient être un sujet de perspectives pour améliorer les démarches collaboratives.*

Perspective 5 *La méthodologie est un exemple concret qui pourrait à l'avenir inspirer de futurs référentiels ou normes sur les interactions multidisciplinaires.*

3. L'EXPERIMENTATION SUR LE CAS D'ETUDE

L'application de la méthodologie sur le cas d'étude a montré la faisabilité et l'effort à fournir pour supporter la synchronisation de modèles entre l'architecture système et la sûreté de fonctionnement. Au total, cinq points de synchronisation ont été appliqués, qui ont permis d'identifier 57 relations de cohérence et 27 incohérences. Ces résultats sont des informations importantes, précises et riches pour les ingénieurs et les architectes.

L'expérimentation a permis de faire des observations sur la méthodologie :

- Elle montre que la mise en cohérence permet d'engager un dialogue entre les expertises, ce qui n'a quasiment jamais été fait en entreprise.
- Si l'application est menée à terme, elle garantit un niveau de cohérence entre les contenus des modèles.
- La réflexion menée durant la mise en cohérence permet d'enrichir les modèles des disciplines d'ingénierie.
- La méthodologie est construite de sorte que les experts n'ont pas besoin de comprendre le formalisme d'une autre discipline. Il leur est cependant nécessaire de comprendre des représentations plus abstraites que les modèles qu'ils manipulent au quotidien.

L'expérimentation faite de la méthodologie présente également des limites. Pour le moment, un seul cas d'étude a été testé, dans le secteur aéronautique. De plus, les modèles du cas d'études sont construits théoriquement par deux disciplines d'ingénierie cependant en pratique un seul individu les a décrits.

Pour que l'expérimentation soit validée, les perspectives 6 et 7 sont envisagées :

Perspective 6 *Pour aller plus loin, de nouveaux cas d'étude dans des secteurs d'activités variés doivent être expérimentés avec la méthodologie.*

Perspective 7 *Pour aller plus loin, d'autres cas d'étude produits par plusieurs individus incarnant des parties prenantes distinctes doivent être étudiés.*

4. TRAVAUX D'IMPLEMENTATION

A terme, une chaîne outillée de synchronisation de modèles serait souhaitée. L'état actuel des travaux montre la faisabilité technologique d'un tel outil. Nous avons pu montrer la possibilité de réaliser des abstractions, des concrétisations et des profils pour la définition des contextes d'ingénierie et des besoins de synchronisation.

L'état actuel et les limites d'implémentation sont :

- Cinq points de vue ont été définis cependant un seul a été implémenté.
- Les deux algorithmes de comparaison n'ont pas été implémentés. Ils ont été utilisés, à plusieurs reprises, manuellement, sur le cas d'étude.
- Le chainage des outils de synchronisation n'a pas été présenté dans le manuscrit. La synchronisation ne présente pas de difficulté d'implémentation. Des expérimentations ont été faites durant ma première année de thèse sur l'environnement Sophia [11]. En exécutant différents outils, nous avons successivement produit des modèles d'un système en AltaRica 3.0 puis en OpenPSA et avons obtenu des résultats de calculs sur la fiabilité du système.

Les travaux d'implémentation ouvrent les perspectives suivantes :

Perspective 8 *Une évaluation des outils de comparaison de modèles permettrait d'implémenter une partie de la synchronisation. Il existe déjà des outils qui permettent d'effectuer des comparaisons de modèles.*

Perspective 9 *Implémenter plus de points de vue et les intégrer permettrait de mettre en œuvre une démarche outillée pour la configuration puis l'application de la synchronisation.*

Perspective 10 *Implémenter des outils de gestion de la traçabilité pourrait permettre de suivre les traces en temps réel et d'obtenir une meilleure visibilité des projets.*

Perspective 11 *Le développement traite de langage UML* [26]*, SysML* [2]*, AltaRica 3.0* [70], [71] *et des structures de données simples mais n'exploite pas d'autres formalismes pour le moment. Il serait pertinent d'appliquer ce type de démarche avec d'autres formalismes tels que Modelica* [140]*, SAML* [93]*, EAST-ADL* [41, 3]*, AADL* [5] *ou Lustre.*

En résumé de la conclusion, les contributions faites sur la synchronisation de modèles apportent des premiers moyens collaboratifs pour une mise en cohérence des modèles utilisés en conception d'architecture système et en sûreté de fonctionnement, tout au long d'un cycle de conception de systèmes complexes.

Au niveau industriel, la modélisation classique en sûreté de fonctionnement utilisant des arbres de défaillance, des blocs diagrammes de fiabilité, etc. restera encore longtemps sur le devant de la scène. Les travaux de thèse contribueront à plus long terme en industrie, lorsque les démarches de modélisation seront basées sur des approches et des modèles plus sophistiqués que celles actuellement utilisées. Enfin, une évolution des pratiques vers plus d'interactivité entre les disciplines dans les phases amont de projet pourrait bénéficier pleinement des réflexions issues de ce travail.

REFERENCES BIBLIOGRAPHIQUES

Bibliographie

[1] J.-M. Jézéquel, B. Combemale et D. Vojtisek, Ingénierie Dirigée par les Modèles : des concepts à la pratique..., Ellipses, Éd., Ellipses, 2012, p. 144.

[2] Object Management Group, *Systems Modeling Language (OMG SysML), formal/2015-06-03,* 2015.

[3] P. Cuenot, P. Frey, R. Johansson, H. Lönn, Y. Papadopoulos, M.-O. Reiser, A. Sandberg, D. Servat, R. T. Kolagari, M. Törngren et e. al., «The EAST-ADL Architecture Description Language for Automotive Embedded Software.,» chez *Model-Based Engineering of Embedded Real-Time Systems*, 2007.

[4] T. N. Qureshi, D. Chen, H. Lönn et M. Törngren, «From EAST-ADL to AUTOSAR Software Architecture: A Mapping Scheme,» chez *Proceedings of the 5th European Conference on Software Architecture*, Berlin, Heidelberg, 2011.

[5] P. H. Feiler et D. P. Gluch, Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language, 1st éd., Addison-Wesley Professional, 2012.

[6] F. Boulanger, «Models, Systems, Heterogeneity, HDR Université Paris Sud - Paris XI,,» Paris XI, 2011.

[7] A. Rauzy, «Presentation : Five Theses for Model-Based Systems Engineering and Model-Based Safety Assessment,» chez *Chaire Blériot-Fabre*, 2016.

[8] ISO/IEC, *ISO-42010 Systems and software engineering - Architecture description,* 2011, pp. 1-46.

[9] A. Villemeur, Sûreté de fonctionnement des système industirels, vol. 67, Eyrolles, 1988.

[10] C. Gomes, B. Barroca et V. Amaral, «Classification of Model Transformation Tools: Pattern Matching Techniques,» *Model-Driven Engineering Languages and Systems,* vol. 8767, pp. 619-635, 2014.

[11] N. Yakymets, S. Dhouib, H. Jaber et A. Lanusse, «Model-driven safety assessment of robotic systems,» *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on,* vol. 1, pp. 1137-1142, November 2013.

[12] D. KROB, CESAMES Systems Architecting Method, A Pocket Guide, CESAMES, Éd., C.E.S.A.M.E.S, 2017.

[13] Map Système, *Pourquoi l'Ingénierie de Système? Un peu d'histoire, http://cluster010.ovh.net/~mapsyste/fr/passe-present-futur/,* Element d'archive.

[14] INCOSE, *Systems Engineering Vision 2020, sebokwiki.org/wiki/INCOSE_Systems_Engineering_Vision_2020,* 2007.

[15] J. Wiley et Sons, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities,* 2015.

[16] Collectif AFIS, «DÉCOUVRIR ET COMPRENDRE L'INGÉNIERIE SYSTÈME, chapitre 11: l'ingénierie système : définition,» *Collection AFIS,* n° %11005, pp. 189-190, 2012.

[17] Object Management Group, *OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1, www.omg.org/spec/UML/2.4.1,* 2011.

[18] Object Management Group, «OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2, www.omg.org/spec/UML/2.1.2/Infrastructure/PDF,» 2007.

[19] Friedrich von HAYEK, traduit par Alain BOYER, *LA THEORIE DES PHENOMENES COMPLEXES, http://www.institutcoppet.org/wp-content/uploads/2011/07/La-th%C3%A9orie-des-ph%C3%A9nom%C3%A8nes-complexes.pdf,* vol. 13, I. Coppet, Éd., 2011.

[20] J. B. Holbrook, «What is interdisciplinary communication? Reflections on the very idea of disciplinary integration,» *Synthese,* vol. 190, pp. 1865-1879, Jul 2013.

[21] SAEAerospace, *ARP4761 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment,* Warrendale, USA, 1996.

[22] ISO, *26262 Road vehicles -- Functional safety,* ISO, Geneva, Switzerland, 2011.

[23] K. E. Kurbel, «The Making of Information Systems: Software Engineering and Management in a Globalized World,» *Springer Verlag,* n° %1ISBN 9783540792604, 2008 .

[24] L. von Bertalanffy, «Théorie générale des systémes, trad. par Jean-Benoist Chabrol,» *Revue Philosophique de Louvain,* vol. 78, n° %137, pp. 161-162, 1980.

[25] Collectif AFIS, «DÉCOUVRIR ET COMPRENDRE L'INGÉNIERIE SYSTÈME, Conception des architectures: chapitre 8, le système est le résultat d'une conception,» *Collection AFIS,* n° %11005, pp. 135-155, 235-249, 250-258, 2012.

[26] Object Management Group, *Unified Modeling Language (OMG UML), formal/2015-03-01,* 2015.

[27] Object Management Group, *Business Process Model and Notation (BPMN) V2.0, formal/2011-01-03,* 2011.

[28] Volvo Technology Corporation, Centro Ricerche Fiat, Continental Automotive Delphi/Mecel Arccore MetaCase Systemite Commissariat a l'Energie Atomique, *MAENAD an FP7 Project supported by the European Commission,* http://www.maenad.eu/, Éd., 2014.

[29] E.-Y. Kang et P.-Y. Schobbens, «Enabling Formal Analysis of Energy-aware Automotive Embedded Systems in East-adl,» chez *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative M&S Symposium*, San Diego, CA, USA, 2013.

[30] S. Moreau, *Principe de la programmation orientée objet, http://lastethese.free.fr/node121.html,* 2003.

[31] ISO, *ISO/IEC/IEEE DIS 42020 Status - Enterprise, systems and software - Architecture processes.*

[32] ISO, *ISO/IEC/IEEE DIS 42030 Status - Enterprise, systems and software - Architecture evaluation.*

[33] ISO/IEC, *ISO 1220-2005 - IEEE Standard for Application and Management of the Systems Engineering Process,* vol. 39, Los Alamitos, CA, USA: IEEE Computer Society, 2011, pp. 92-94.

[34] GEIA/EIA, *EIA-632 Processes for Engineering a System,* 2003.

[35] ISO/IEC, *ISO/IEC/IEEE 15288 Systems and Software Engineering - System Life Cycle Processes,* 2008.

[36] I. Malavolta, P. Lago, H. Muccini, P. Pellicione et A. Tang, *Architectural Languages BenchMark,* 2011.

[37] P. Roques, «MBSE with the ARCADIA Method and the Capella Tool,» chez *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, Toulouse, France, 2016.

[38] Object Management Group, «Software Process Engineering Metamodell SPEM 2.0 OMG Draft Adopted Specification,» 2006.

[39] Object Management Group, *OMG Object Constraint Language (OCL), Version 2.3.1, www.omg.org/spec/OCL/2.3.1/,* 2012.

[40] Object Management Group, *OMG Systems Modeling Language (OMG SysML™),* 2012.

[41] H. Lönn, T. Saxena, M. Sjödin et M. Törngren, «FAR EAST: Modeling an Automotive Software Architecture Using the EAST ADL,» chez *ICSE 2004 workshop on Software Engineering for Automotive Systems (SEAS)*, 2004.

[42] Schneider Electric, «La Sûreté de Fonctionnement - revue InterSections oct 2004,» *InterSections,* vol. 1, n° %11, pp. 1-12, Nov 2004.

[43] E. Seedhouse, «Project MX981,» chez *Pulling G: Human Responses to High and Low Gravity*, New York, NY, Springer New York, 2013, pp. 1-21.

[44] B. Frank, *LA CATASTROPHE DE MALPASSET EN 1959,* 2002.

[45] Departement of Defense of US, MIL HDBK 217 F, DoD, Éd., DOD, 1991.

[46] GSIEN Fiche technique n°30, *La Gazette Nucléaire n°28, Le rapport Rasmussen, WASH-1400,* 1979.

[47] C. C. for chemical process safety, *Il y a 30 ans : la catastrophe de Mexico,* 2014.

[48] G. Point et A. Rauzy, «AltaRica : Constraint automata as a description language,» *Journal européen des systèmes automatisés,* vol. vol.33, n° %18-9, pp. 1033-1052, 1999.

[49] Union technique de l'électricité, UTE C 80-810, AFNOR éd., vol. 1, RDF, 2005.

[50] H. Procaccia et J. Procaccia, Genèse du Big Bang à l'art pariétal, 2. Saint Denis : Connaissances et savoirs, Éd., Sciences de l'univers, 2015.

[51] NF, Norme Française, *Norme NF X60-500 Terminologie relative à la fiabilité, maintenabilité, disponibilité,* 1988.

[52] M. Sayed Mouchaweh, «Diagnostic des systèmes à Evènements Discrets (SED) Etat de l'art,» *Techniques de l'ingénieur Méthodes de production,* vol. base documentaire : TIB521DUO., n° %1ref. article : ag3540, 2011.

[53] J.-C. Laprie, Guide de la sûreté de fonctionnement, 2ème édition éd., Cépaduès, Éd., Broché, 1996.

[54] A. Avizienis, J.-C. Laprie, B. Randell et C. Landwehr, «Basic Concepts and Taxonomy of Dependable and Secure Computing,» *IEEE Trans. Dependable Secur. Comput.,* vol. 1, pp. 11-33, Jan 2004.

[55] C. Pagetti, «Module de sûreté de fonctionnement, dspace.univ-tlemcen.dz/bitstream/112/9395/1/cours.pdf,» 2012.

[56] Y. Mortureux, «La sureté de fonctionnement : methodes pour maîtriser des risques,» *Techniques de l'ingénieur méthodes d'analyse des risques,* vol. TIB155DUO., n° %1ag4670, oct 2001.

[57] NF, Norme Française, *Norme NF X60-151 à 156 Analyse de la valeur MISME,* 1991.

[58] Case France, *Analyse et conception SART, http://www.case-france.com/EnvisionSART.html,* Element d'archive.

[59] AFNOR, *NF EN 31010 Gestion des risques - Techniques d'évaluation des risques.*

[60] M. ROYER, *HAZOP : une méthode d'analyse des risques- Présentation et contexte,* 2016.

[61] P. PERILHON, *MOSAR- Présentation de la méthode,* 2003.

[62] AFNOR, *NF EN 60812 Techniques d'analyses de la fiabilité du système - Procédure d'analyse des modes de défaillance et de leurs effets (AMDE).*

[63] G. Landy, AMDEC guide pratique, Afnor, Éd., Afnor Edition, 2011.

[64] IEEE , «IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software,» *IEEE Std 982.2-1988,* n° %1ISBN: 0-7381-0398-5, 1989.

[65] IRNS, La méthode de l'arbre des causes, L'analyse de l'accident du travail, vol. ED 6163, Brochure, Éd., IRNS, 2013.

[66] F. Lees, Lees' Loss Prevention in the Process Industries, vol. Vol 3, ELSEVIER, Éd., Butterworth-Heinemann, 2004.

[67] Dependability, TC 56, *IEC 61025 Fault tree analysis,* 2006.

[68] AFNOR, *NF EN 62551 Techniques d'analyse de sûreté de fonctionnement - Techniques des réseaux de Pétri,* 2013.

[69] J.-P. SIGNORET, «Analyse des risques des systèmes dynamiques : approche markovienne,» *Techniques de l'ingénieur,* vol. RÉF : SE4071 V1, p. 26, Oct 2005.

[70] T. Prosvirnova, «AltaRica 3.0: a Model-Based approach for Safety Analyses,» 2014.

[71] T. Prosvirnova et A. Rauzy, «The structural constructions of AltaRica 3.0,» chez *Actes du congrès LambdaMu19 (actes électroniques)*, Dijon (France), 2014.

[72] ISO/IEC, *ISO/IEC 61508 Functional safety of electrical/electronic/programmable electronic safety-related,* 1990.

[73] N. Yakymets, M. Perin et A. Lanusse, «Model-Driven Multi-Level Safety Analysis of Critical Systems,» chez *SysCon 2015 / TFSE , Vancouver, British Columbia, Canada, April 13-16*, 2015.

[74] N. Yakymets, H. Jaber et A. Lanusse, «Model-based System Engineering for Fault Tree Generation and Analysis,» chez *MODELSWARD 2013 - Proceedings of the 1st International Conference on Model-Driven Engineering and Software Development, Barcelona, Spain, 19 - 21 February, 2013*, 2013.

[75] T. Prosvirnova, «AltaRica 3.0: a Model-Based approach for Safety Analyses,» 2014.

[76] M. B. Batteux, T. Prosvirnova et A. Rauzy, «System Structure Modeling Language (S2ML),» 2015.

[77] A. Rauzy, «Guarded transition systems: A new states/events formalism for reliability studies,» *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability,* vol. 222, n° %14, pp. 495-505, 2008.

[78] M. Bozzano et A. Villafiorita, «Improving System Reliability via Model Checking: The FSAP/NuSMV-SA Safety Analysis Platform,» chez *Computer Safety, Reliability, and Security: 22nd International Conference, SAFECOMP 2003, Edinburgh, UK, September 23-26, 2003. Proceedings*, S. Anderson, M. Felici et B. Littlewood, Éds., Berlin, Heidelberg, Springer Berlin Heidelberg, 2003, pp. 49-62.

[79] B. Bittner, M. Bozzano, R. Cavada, A. Cimatti, M. Gario, A. Griggio, C. Mattarei, A. Micheli et G. Zampedri, «The xSAP Safety Analysis Platform,» *CoRR,* vol. abs/1504.07513, n° %12, pp. p1-9, Avr 2015.

[80] M. Bouissou, *Automated Dependability Analysis of Complex Systems with the KB3 Workbench: the Experience of EDF R&D,* 2005.

[81] M. Adachi, Y. Papadopoulos, S. Sharvia, D. Parker et T. Tohdo, «An approach to optimization of fault tolerant architectures using HiP-HOPS,» *Softw., Pract. Exper.,* Vols. %1

sur %241, n°11, pp. 1303-1327, 2011.

[82] J. B. Dugan et al., «DIFTree: a software package for the analysis of dynamic fault tree models,» Philadelphia, 1997.

[83] R. Cressent, V. Idasiak et F. Kratz, «Rapprocher les études de sûreté de fonctionnement de l'ingènierie système : retour d'expérience,» chez *QUALITA 2011*, France, 2011.

[84] P. Mauborgne, «Vers une ingénierie de systémes sûrs de fonctionnement basée sur les modèles en conception innovante,» 2016.

[85] P. Mauborgne, S. Deniaud, É. Levrat, É. Bonjour, J.-P. Micaëlli et D. Loise, «The Determination of Functional Safety Concept coupled with the definition of Logical Architecture: a framework of analysis from the automotive industry,» *IFAC PapersOnLine 50-1,* vol. 50, n° %11, pp. 7278-7283, 2017.

[86] P. Mauborgne, D. S., É. Levrat, É. Bonjour, J.-P. Micaëlli et D. Loise, «Operational and System Hazard Analysis in a Safe Systems Requirement Engineering Process – Application to automotive industry,» *Safety Science,* vol. 87, pp. 256-268, 2016.

[87] R. Cressent, V. Idasiak, F. Kratz et P. David, «Mastering Safety and Reliability in a Model Based Process,» chez *2011 Proceedings - Annual Reliability and Maintainability Symposium*, Lake Buena Vista, FL, United States, 2011.

[88] R. Cressent, V. Idasiak et F. Kratz, «Prise en compte des analyses de la sûreté de fonctionnement dans l'ingénierie de système dirigèe par les modèles SysML,» *Génie logiciel,* vol. 1, pp. p33-39, Mar 2011.

[89] R. T. Kolagari, D. Chen, A. Lanusse, R. Librino, H. Lönn, N. Mahmud, C. Mraidha, M.-O. Reiser, S. Torchiaro, S. Tucci-Piergiovanni, T. Wägemann et N. Yakymets, «Model-Based Analysis and Engineering of Automotive Architectures with EAST-ADL: Revisited,» *Int. J. Concept. Struct. Smart Appl.,* vol. 3, n° %12, pp. 25-70, Jul 2015.

[90] S. Tucci-Piergiovanni, D. Chen, C. Mraidha, H. Lönn, N. Mahmud, M.-O. Reiser, R. Tavakoli Kolagari, N. Yakymets, R. Librino et S. Torchiaro, «Model-Based Analysis andEngineering of AutomotiveArchitectures with EAST-ADL,» chez *Handbook of Research on Embedded Systems Design*, 2014, pp. 242-282.

[91] M. Bozzano, A. Cimatti, J.-P. Katoen, V. Y. Nguyen, T. Noll et M. Roveri, «Safety, Dependability and Performance Analysis of Extended AADL Models,» *Comput. J.,* vol. 54, n° %15, pp. 754-775, Mai 2011.

[92] «Model-Based Safety and Assessment - 5th International Symposium, IMBSA,» 2017.

[93] «Model-Based Safety and Assessment - 4th International Symposium, IMBSA 2014, Munich, Germany, October 27-29, 2014. Proceedings,» 2014.

[94] C. Guychard, S. Guerin, A. Koudri, A. Beugnard et F. Dagnat, «Conceptual interoperability through Models Federation,» chez *Semantic Information Federation Community Workshop*, Miami, United States, 2013.

[95] L. Wouters, Y. Kaeri et K. Sugawara, «Multi-domain multi-lingual collaborative design,» chez *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2013.

[96] G. Oster, P. Molli, H. Skaf-Molli et A. Imine, «Un modèle sûr et générique pour la synchronisation de données divergentes,» chez *Premières Journées Francophones : Mobilité et Ubiquité - UbiMob'04*, Nice, France, 2004.

[97] M. Abilov, T. Mahmoud, J. M. Gómez et M. Mora, «Towards an Incremental Bidirectional Partial Model Synchronization between Organizational and Functional Requirements Models,» chez *Proc.\ MoDRE*, 2015.

[98] P. Arnold et E. Rahm, «Semantic Enrichment of Ontology Mappings: A Linguistic-Based Approach,» chez *Advances in Databases and Information Systems: 17th East European Conference, ADBIS 2013, Genoa, Italy, September 1-4, 2013. Proceedings*, B. Catania, G. Guerrini et J. Pokorny, Éds., Berlin, Heidelberg, Springer Berlin Heidelberg, 2013, pp. 42-55.

[99] M. Perin et L. Wouters, «Using Ontologies for Solving Cross-Domain Collaboration Issues,» *IFAC Proceedings Volumes,* vol. 47, n° %13, pp. 7837-7842, 2014.

[100] G. Kappel, E. Kapsammer, H. Kargl, G. Kramler, T. Reiter, W. Retschitzegger et W. W. M. Schwinger, «Lifting Metamodels to Ontologies: A Step to the Semantic Integration of Modeling Languages,» chez *Model Driven Engineering Languages and Systems: 9th International Conference, MoDELS 2006, Genova, Italy, October 1-6, 2006. Proceedings*, O. Nierstrasz, J. Whittle, D. Harel et G. Reggio, Éds., Berlin, Heidelberg, Springer Berlin Heidelberg, 2006, pp. 528-542.

[101] D. Torre, Y. Labiche et M. Genero, «UML Consistency Rules: A Systematic Mapping Study,» chez *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, New York, NY, USA, 2014.

[102] M. Didonet Del Fabro, J. Bézivin, F. Jouault, E. Breton et G. Gueltas, «AMW: a generic model weaver,» chez *1ere Journées sur l'Ingénierie Dirigée par les Modèles (IDM05)*, France, 2005.

[103] V. Haren, TOGAF Version 9.1, 10th éd., Van Haren Publishing, 2011.

[104] KLM, *PFD Process flow diagrams (Project standards and specifications),* Jalan Sri Perkasa 2 Taman Tampoi Utama 81200 Johor Bahru Malaysia, 2011.

[105] I. Rosziati et Y. y. Siow, «Formalization of the Data Flow Diagram Rules for Consistency Check,» *International Journal of Software Engineering & Applications (IJSEA),* vol. 1, Oct 2010.

[106] ISO, *ISO 9000 Management de la Qualité,* 2015.

[107] AFNOR, *NF EN 1325-1 Vocabulaire du management de la valeur, de l'analyse de la valeur et de l'analyse fonctionnelle - Partie 1 : analyse de la valeur et analyse fonctionnelle.,* 1996.

[108] ISO/IEC, *NF X50-100 Management par la valeur - Analyse fonctionnelle, caractéristiques fondamentales - Analyse fonctionnelle : analyse fonctionnelle du besoin (ou externe) et analyse fonctionnelle technique/produit (ou interne) - Exigences sur les livrables et*

démarches de mise en oeuvre, 2011.

[109] ISO/IEC, *ISO 21351 Systèmes spatiaux -- Cahier des charges fonctionnel et spécification technique de besoin,* 2005.

[110] AFNOR, *NF X50-120 Vocabulaire pour le management et l'assurance de la qualité,* 1987.

[111] A. Rauzy, *An implementation of model-based safety assessment- Keynote Esrel 2017,* 2017.

[112] J. Bézivin, «Sur les principes de base de l'ingénierie des modèles.,» *L'OBJET,* Vols. %1 sur %210, n°4, pp. 145-157, 2004.

[113] K. Czarnecki et S. Helsen, «Classification of Model Transformation Approaches,» chez *OOPSLA'03 Workshop on Generative Techniques in the Context of Model-Driven Architecture*, 2003.

[114] E. Kindler et R. Wagner, «Triple Graph Grammars: Concepts, Extensions, Implementations, and Application Scenarios,» 2007.

[115] G. Szàrnyas, B. Izso, I. Ràth, D. Harmath, G. Bergmann et D. Varro, «IncQuery-D: A Distributed Incremental Model Query Framework in the Cloud,» chez *Model-Driven Engineering Languages and Systems: 17th International Conference, MODELS 2014, Valencia, Spain, September 28 -- October 3, 2014. Proceedings*, J. Dingel, W. Schulte, I. Ramos, S. Abrahäo et E. Insfran, Éds., Cham, Springer International Publishing, 2014, pp. 653-669.

[116] Z. Ujhelyi, G. Bergmann, A. Hegedüs, A. Horvàth, B. Izso, I. Ràth, Z. Szatmàri et D. Varro, «EMF-IncQuery: An integrated development environment for live model queries,» *Sci. Comput. Program.,* vol. 98, pp. 80-99, 2015.

[117] Object Management Group, *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, Version 1.1, www.omg.org/spec/QVT/1.1/,* 2011.

[118] I. Kurtev, «State of the Art of QVT: A Model Transformation Language Standard,» chez *Applications of Graph Transformations with Industrial Relevance: Third International Symposium, AGTIVE 2007, Kassel, Germany, October 10-12, 2007, Revised Selected and Invited Papers*, A. a. N. M. a. Z. A. Schürr, Éd., Berlin, Heidelberg, Springer Berlin Heidelberg, 2008, pp. 377-393.

[119] F. Jouault et I. Kurtev, «Transforming Models with ATL,» chez *Satellite Events at the MoDELS 2005 Conference: MoDELS 2005 International Workshops Doctoral Symposium, Educators Symposium Montego Bay, Jamaica, October 2-7, 2005 Revised Selected Papers*, J. Bruel, Éd., Berlin, Heidelberg, Springer Berlin Heidelberg, 2006, pp. 128-138.

[120] S. Eugene, V. Hans, M. Raphael, H. Conner, V. M. Simon et E. Hüseyin, «AToMPM: A Web-based Modeling Environment,» chez *Joint Proceedings of MODELS'13 Invited Talks, Demonstration Session, Poster Session, and ACM Student Research Competition co-located with the 16th International Conference on Model Driven Engineering Languages and Systems (MODELS 2013), Miami, USA, September 29 - October 4, 2013.*, 2013.

[121] J.-L. Viruéga, Traçabilité (outils, méthodes, pratiques), Eyrolles éd., vol. 1, E.

d'Organisation, Éd., Eyrolles, 2005.

[122] Concepts et t. commity, *ISO 8402 Quality management and quality assurance -- Vocabulary, www.iso.org/standard/20115.html,* 1994.

[123] ISO/TC 176/SC 2 Systèmes qualité, *ISO 9001 Systèmes de management de la qualité - Exigences,* 2015.

[124] ISO/TC 176/SC 2 Systèmes qualité, *ISO 9002 Systèmes qualité - Modèle pour l'assurance de la qualité en production, installation et prestations associées,* 1994.

[125] B. Faraggi, Traçabilité (French Edition), DUNOD éd., vol. 1, L. Nouvelle, Éd., DUNOD, 2006.

[126] A. Saïd et E.-G. Hamid, «Une approche générique pour la définition de modèles de traçabilité,» *Journée commune IDM et INFORSID,* p. 1, Novembre 2009.

[127] A. De Varro, «Jean Lacroix, Marxisme, existentialisme, personnalisme,» *Revue Philosophique de Louvain,* vol. 48, n° %118, pp. 302-303, 1950.

[128] Departement of Defense, The DoDAF Architecture Framework Version 2.02, US. Department of Defense, 2010.

[129] I. Bailey, «Brief Introduction to MODAF with v1.2 Updates,» chez *2008 IET Seminar on Enterprise Architecture Frameworks*, 2008.

[130] NATO, North Atlantic Treaty Organization, «NATO Architecture Framework Version 3.0 (NAF V2.0),» 2007.

[131] SAEAerospace, *ARP4754 Certification Considerations for Highly-Integrated Or Complex Aircraft Systems,* Warrendale, USA, 2010.

[132] B. Brosgol, «DO-178C: The Next Avionics Safety Standard,» *Ada Lett.,* vol. 31, pp. 5-6, Nov 2011.

[133] RTCA, *DO-178C Software Considerations in Airborne Systems and Equipment Certification,* 2006.

[134] S. Gérard, C. Dumoulin, P. Tessier et B. Selic, «Papyrus: A UML2 Tool for Domain-specific Language Modeling,» chez *Proceedings of the 2007 International Dagstuhl Conference on Model-based Engineering of Embedded Real-time Systems*, Berlin, Heidelberg, 2010.

[135] S. Epstein et A. Rauzy, *Open-PSA Model Exchange Format,* 2008.

[136] E. D. Francesco, R. D. Francesco et E. Petritoli, «Obsolescence of the MIL-HDBK-217: A critical review,» chez *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2017.

[137] A. Rauzy, «XFTA An Open-PSA FaultTree Engine, altarica-association.org/downloads/rams/xfta/XFTA-Manual.pdf,» 2012.

[138] T. Prosvirnova, E. Saez, C. Seguin et P. Virelizier, «Handling Consistency between safety analysis and system models,» *Model-Based Safey and Assessment,* vol. 10437, pp. 19-34,

Sep 2017.

[139] Object Management Group, *Meta-Object Facility (MOF) Specification, Version 2.0,* 2006.

[140] I. Bumin Kara, «Design and Implementation of the ModelicaML Code Generator Using Acceleo 3.X,» Department of Computer and Information Science, 2015.

[141] A. Legendre, *Frise chronologique des évolutions des études en sûreté de fonctionnement,* 2014.

[142] Communauté Web, *Architecture (informatique) - Wikipédia, https://fr.wikipedia.org/wiki/Architecture_(informatique).*

WEBOGRAPHIE

[web1] The MathWorks, Inc.. *"Simulink Tool",* sur le site The MathWorks, Inc.. Consulté le 19 septembre 2017.
https://fr.mathworks.com/products/simulink.html

[web2] Modelica Association. *« Modelica Language »,* sur le site Modelica Association. Consulté le 31 août 2017.
https ://www.modelica.org

[web3] Ledfrod, Heidi (2015). *"How to solve the world's biggest problems",* sur le site Nature. Consulté le 31 août 2017.
https ://www.nature.com/news/how-to-solve-the-world-s-biggest-problems-1.18367

[web4] Philippe Berger. *« Analyse S.A.D.T. »,* sur le site Page personnelle de Philippe Berger sur free.fr. Consulté le 31 août 2017.
http://philippe.berger2.free.fr/automatique/cours/sadt/sadt.htm#top

[web5] Case France. *« Analyse et conception SART »,* sur le site Case France. Consulté le 31 août 2017.
http ://www.case-france.com/EnvisionSART.html

[web6] Wikimedia Foundation (2010). *« Comparaison des logiciels UML »,* sur le site Fracademic. Consulté le 31 août 2017.
http ://fracademic.com/dic.nsf/frwiki/1888634

[web7] No Magic. *"Cameo Systems Modeler Tool ",* sur le site No Magic. Consulté le 31 août 2017.
https ://www.nomagic.com/products/cameo-systems-modeler#features

[web8] Sparx Systems. *"Enterprise Architect Tool ",* sur le site Sparx Systems. Consulté le 31 août 2017.
https ://www.sparxsystems.com/products/mdg/tech/sysml/index.html

[web9] ModelioSoft. *« Modelio SA Tool »,* sur le site ModelioSoft. Consulté le 31 août 2017.
https ://www.modeliosoft.com/en/products/modelio-sa-overview.html

[web10] IBM. *"IBM Rhapsody Tool"*, sur le site IBM. Consulté le 31 août 2017.
http ://www-03.ibm.com/software/products/en/ratirhaparchforsystengi

[web11] PTC. *« ARTiSAN Studio Tool »,* sur le site PTC. Consulté le 31 août 2017.
https ://www.ptc.com/en/model-based-systems-engineering/integrity-modeler

[web12] OBEO. *« Capella Tool »*, sur le site Polarsys. Consulté le 31 août 2017.
https ://www.polarsys.org/capella

[web13] CEA LIST. *"Papyrus Tool",* sur le site Polarsys. Consulté le 31 août 2017.
https://www.polarsys.org/solutions/papyrus

[web14] Anthony Legendre. "Frise chronologique des évolutions des études en sûreté de

fonctionnement", sur le site Timeglider. Consulté le 22 septembre 2015. http://fresques.ina.fr/jalons/fiche-media/InaEdu01039/le-naufrage-du-petrolier-torrey-canyon.html

[web15] *"Chronologie de catastrophes industrielles ",* sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Chronologie_de_catastrophes_industrielles

[web16] *"Liste des accidents ferroviaires en France au xixe siècle",* sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Liste_des_accidents_ferroviaires_en_France_au_XIXe_si%C3%A8cle

[web17] *"Naufrage du Titanic",* sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Naufrage_du_Titanic

[web18] *"Arbre de défaillances Historique",* sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Arbre_de_d%C3%A9faillances#Historique

[web19] *"Barrage de Vajont",* sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Barrage_de_Vajont

[web20] *"Accident de l'USS Forrestal "*, sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Accident_de_l%27USS_Forrestal

[web21] *INA. "Le naufrage du pétrolier Torrey Canyon", sur le site Jalons Version découverte. Consulté le 31 août2017. http://fresques.ina.fr/jalons/fiche-media/InaEdu01039/le-naufrage-du-petrolier-torrey-canyon.html*

[web22] *"Barrage de Banqiao",* sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Barrage_de_Banqiao

[web23] *"Catastrophe de Morvi",* sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Catastrophe_de_Morvi

[web24] "*Accident nucléaire de Three Mile Island* ", sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Accident_nucl%C3%A9aire_de_Three_Mile_Island

[web25] *"Essais nucléaires français ",* sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Essais_nucl%C3%A9aires_fran%C3%A7ais

[web26] *"Cercle de qualité"*, sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Cercle_de_qualit%C3%A9

[web27] *"Catastrophe nucléaire de Tchernobyl",* sur le site Wikipédia. Consulté le 1 septembre 2017. https://fr.wikipedia.org/wiki/Catastrophe_nucl%C3%A9aire_de_Tchernobyl

[web28] *"Accident de la navette spatiale Challenger",* sur le site Wikipédia. Consulté le 1

septembre 2017.
https://fr.wikipedia.org/wiki/Accident_de_la_navette_spatiale_Challenger

[web29] AV Canada (2015). *"LA MÉTHODE FAST (FONCTION ANALYSIS SYSTEM TECHNIQUE)"*, sur le site AV Canada. Consulté le 31 août 2017.
http://scav-csva.org/fast.php?lang=fr

[web30] ESPINASSE, Bernard. *"MERISE : une méthode systèmique de conception de SI"*, sur le site Laboratoire des sciences de l'information et des systèmes. Consulté le 31 août 2017.
http://www.lsis.org/dea/M6optionD/Exp-GL5Merise.pdf

[web31] Lloyd's Register group*. "Risk Spectrum Tool",* sur le site Riskspectrum. Consulté le 31 août 2017.
http://www.riskspectrum.com/en/risk/

[web32] Apsys Airbus. *"Simfia V3 Tool",* sur le site Apsys Airbus. Consulté le 31 août 2017.
https://www.apsys-airbus.com/digital-software/#SIMFIA

[web33] Satodev. *"Grif Tool ",* sur le site Satodev. Consulté le 31 août 2017.
http://grif-workshop.fr

[web34] Labri. *"Altarica Project, History and dialects",* sur le site Labri. Consulté le 31 août 2017.
https://altarica.labri.fr/wp/?page_id=23

[web35] IRT System X. *"OpenAltaRica Tool "*, sur le site IRT System X. Consulté le 31 août 2017.
http://openaltarica.fr

[web36] Entretien avec Bruno Bachimont (Juillet 2006). *"Qu'est-ce qu'une ontologie?"*, sur le site Technolangue. Consulté le 31 août 2017.
http://www.technolangue.net/imprimer.php3?id_article=280

[web37] Frederic Madiot (juin 2017). *"Acceleo Tool",* sur le site Eclipse. Consulté le 19 août 2017.
https://wiki.eclipse.org/Acceleo

[web38] DGA (2014). *"Document de Présentation de l'Orientation de la S&T Période 2014 2019"*, sur le site Ministère des armées, Bagneux. Consulté le 10 août 2017.
https://www.ixarm.com/IMG/pdf/post_dga_2014_2019.pdf

LEXIQUE

LEXIQUE

Dans ce mémoire, plusieurs concepts ont été utilisés. Certains sont définis différemment selon les ouvrages et les standards. Les définitions retenues pour ces concepts sont donc présentées dans le tableau ci-dessous.

Terme	Définition	Renvoie au texte
Abstraction	*L'abstraction est une technique de transformation de modèles. C'est l'activité inverse du raffinement. Les informations d'un modèle sont regroupées ou éliminées pour simplifier le modèle. L'objectif de l'abstraction est de masquer certaines informations pour mettre l'accent sur une préoccupation spécifique. Si le modèle M1 raffine le modèle M2 alors M2 est une abstraction de M1.*	p.64
Activité	*C'est l'application d'une méthode particulière à un instant donné durant un processus ou une méthodologie. Selon ISO 9000 (2005)* [106] *et ISO 15288* [35]*, l'activité définit clairement les éléments d'entrée et de sortie dont la valeur ajoutée est mesurable. Il peut exister des éléments intermédiaires pour représenter des résultats non-finaux. Ainsi, une activité est définie par un objectif de résultat attendu (qui peut être un sous-objectif d'un processus). Il est possible de considérer une condition (une garde) sur la faisabilité de l'activité en fonction de la maturité des éléments d'entrée et/ou de l'accomplissement des activités en amont.*	p.45
Analyse et Evaluation de la sûreté de fonctionnement basée sur les modèles (MBSA)	*C'est l'application formalisée et outillée de la modélisation, permettant à l'ingénieur de sûreté de fonctionnement de gérer ses modèles et ses activités tout au long de son processus. »*	p.29
Architecture	*L'ensemble des concepts fondamentaux et les propriétés d'un système dans son environnement incarné par ses éléments, ses relations et ses principes de conception et d'évolution. ISO 42010 [3] et TOGAF 9.1* [103]	p.43
Architecture description	*C'est le résultat utilisé pour exprimer une architecture. ISO 42010 [3]*	p.43
Architecture système	*Elle est responsable de deux missions.* *La première est de proposer des solutions d'architecture, i.e. définir comment les composantes (fonctions, composants physiques, logiciels, etc.) du système sont organisées et assemblées entre elles. Ces solutions sont contenues dans des modèles à plusieurs niveaux d'abstraction et de raffinement (vision opérationnelle, vision fonctionnelle, vision organique plus ou moins détaillée).* *La seconde mission est d'interagir avec les disciplines techniques et transverses, pour obtenir des retours sur les solutions d'architecture proposées. Elle doit adapter ses représentations selon les préoccupations des disciplines ciblées.*	p.19
Besoin de synchronisation	*C'est une relation entre plusieurs points de vue d'architecture issus de différents contextes. Il est aussi caractérisé comme un instant identifié entre les processus des disciplines d'ingénierie où la mise en cohérence d'informations est possible. Ce besoin répond à un besoin formalisé de partage de points de vue par des disciplines manipulant des objets ayant des dépendances entre elles.*	p.45
Cohérence	*C'est une propriété d'une relation entre des éléments de différentes vues. C'est une équivalence en rapport à un certain sujet, i.e. que les éléments font référence à une même réalité. Elle peut être complètement ou partiellement satisfaite.*	p.60
Comparaison	*La comparaison de modèles est une technique de construction de relations spécifiques entre des éléments de deux modèles. La comparaison est chargée de construire des relations et d'en interpréter des résultats selon des critères. Les modèles comparés doivent être décrits par le même langage.*	p.64
Concept(s) pivot(s) d'architecture	*C'est l'ensemble des concepts structurants communs aux modèles utilisés par les disciplines d'ingénierie. Ils font référence aux sujets soumis à la mise en cohérence lors d'une interaction multidisciplinaire. Ils définissent le métamodèle pivot qui sera utilisé lors d'une itération.*	p.50

Concrétisation	*La concrétisation est une technique de transformation de modèles. La concrétisation est une activité de raffinement de modèle. Des informations sont ajoutées dans un modèle pour le raffiner. L'objectif de la concrétisation est de rendre plus concret le contenu du modèle.*	p.64
Conditions de validation	*C'est l'ensemble de règles de validation qui doit être vérifié avant chaque application de la synchronisation.*	p.56
Contexte d'une discipline d'ingénierie	*C'est une notion abstraite qui inclut toutes les compétences: savoir, savoir-faire et savoir-être spécifiques à une discipline d'ingénierie. Il a un but (ou un ensemble d'objectifs), généralement pour rendre un service, un produit ou un résultat attendu. Le contexte est rattaché à une discipline d'ingénierie.*	p.45
Discipline d'ingénierie	*C'est une partie prenante active. Elle est chargée de répondre à un besoin d'étude d'un système sous certaines préoccupations.*	p.40
Discipline d'ingénierie	*C'est une partie prenante. Elle possède des préoccupations spécifiques à un domaine d'étude ou d'analyse. Elle intervient sur un système durant son cycle de vie pour répondre à des problématiques non triviales.*	p.45
Elément	*C'est une représentation d'un objet dans un modèle. C'est aussi un conteneur, i.e. un package pouvant contenir d'autres éléments ou du comportement, e.g. une fonction, un composant, etc. Les langages proposent des méta-classes telles que : une classe uml, un block Sysml, un* *Et l'ana node en altarica dataflow, une classe ou un block en AltaRica 3.0, etc.*	p.50
Fonction de transformation	*C'est une fonction qui prend des éléments sources particuliers et qui construit un ou plusieurs élément(s) cible(s) équivalent(s) à partir des métamodèles. Elle définit également des gardes et des relations de traçabilité entre ces éléments.*	p.70
Ingénierie système	*C'est un cadre qui englobe l'ensemble des activités des disciplines d'ingénierie. Ce cadre multidisciplinaire définit les périmètres des études des disciplines. Son but est de favoriser la réalisation d'un système performant comme solution finale aux besoins d'un client, tout en satisfaisant les* ***parties prenantes****.*	p.16
Ingénierie système basée sur les modèles	*L'INCOSE définit* [14]*: « L'ingénierie système basée sur les modèles (MBSE), est l'application formalisée de la modélisation permettant la gestion des exigences du système, la conception, l'analyse, les activités de vérification et de validation dès les étapes amont et tout au long du cycle de vie ».*	p.21
Ingénierie système basée sur les modèles (MBSE)	*C'est l'application formalisée de la modélisation permettant à l'architecture système de gérer ses modèles et ses activités tout au long de son processus. »*	p.21
Langage de modélisation spécifique au secteur d'activité (DSML)	*C'est un langage de modélisation (généralement graphique) pour créer des modèles spécifiques à un certain domaine (e.g. diagnostic de maladie, configuration « Quality of Service »). Il offre une expressivité axée sur une problématique particulière grâce à une notation et des abstractions appropriées.*	p.24
Lien de traçabilité	*Liens d'un ensemble d'éléments sources à un ensemble d'éléments cibles.*	p.73
Listes de compromis	*Ce sont des ensembles d'opérations pouvant être appliquées sur les modèles. Chaque discipline d'ingénierie peut proposer d'appliquer un compromis pour tenter de résoudre une incohérence identifiée. Les listes peuvent être limitées par le rôle des disciplines.*	p.56
Livrable	*Résultats/Documents attendus après la réalisation d'une activité. Il contient les résultats de sortie de l'activité dans son contexte.*	p.78
Mapping	*Le mapping est un ensemble ordonné de relations reliant les objets du ou des point(s) de vue d'un contexte vers les objets du métamodèle pivot. La notion de mapping est fortement liée aux principes de transformation de modèles (cf. 6.3 Transformation de modèles).*	p.56
Métamodèle pivot	*C'est la définition des modèles utilisés pour exprimer les concepts communs des vues des disciplines d'ingénierie. Il peut y avoir plusieurs métamodèles pivots dans l'exécution des interactions multidisciplinaires.*	p.50
Méthode	*C'est un ensemble ordonné de manière logique, de principes, de règles, d'étapes qui constituent un moyen pour parvenir à un résultat attendu. Une méthode emploie des éléments d'entrée et produit des éléments de sortie. Ces derniers sont représentés dans des vues graphiques, textuelles, …*	p.45
Méthodologie	*Branche de la logique étudiant les méthodes des différentes sciences* [127]*. C'est un processus générique, qui peut être décliné sous des formes plus spécifiques en un ensemble d'activités séquentiellement organisées.*	p.78
Mission	*Les fonctions ou activités remplies par une discipline d'ingénierie sur un système.*	p.79
Modèle (en ingénierie système)	*C'est une abstraction d'un système réel ou étudié. Il s'appuie sur un cadre mathématique adapté à un ensemble d'objectifs et un ensemble de vues, i.e. des représentations définies par un point de vue.*	p.16

Motivation du sponsor	*L'ensemble des facteurs déterminant le choix du sponsor à sélectionner une discipline d'ingénierie.*	p.79
Objectif du sponsor	*Raison de la sélection des disciplines d'ingénierie par le sponsor pour la mise en œuvre d'une synchronisation.*	p.79
Objectif	*Raison d'usage d'une activité, d'un processus ou d'une méthodologie. L'objectif doit être satisfait par les livrables produits.*	p.78
Partie prenante	*C'est un acteur, individuel ou collectif (groupe ou organisation), activement ou passivement concerné par une décision ou un projet, i.e. dont les intérêts peuvent être affectés positivement ou négativement à la suite de son exécution (ou de sa non-exécution). (« stakeholder »)*	p.16
Point de synchronisation	*C'est une configuration de mise en cohérence de deux points de vue caractérisant trois fonctions de base (l'abstraction, la comparaison et la concrétisation). Il peut avoir des antécédents.*	p.56
Point de vue d'Architecture (Viewpoint)	*C'est un résultat définissant les modalités de construction, d'interprétation et d'utilisation d'une vue d'architecture pour capturer des préoccupations particulières du système. ISO 42010 [3]*	p.43
Préoccupation	*Intérêt principal qui est d'une importance cruciale pour les parties prenantes d'un système et déterminante pour l'acceptabilité du système. Les préoccupations peuvent concerner tous les aspects du fonctionnement, du développement ou de l'exploitation du système, y compris des considérations telles que la performance, la fiabilité, la sécurité, la distribution et l'évolutivité. ISO 42010 [3] et TOGAF 9.1* [103]	p.43
Principe	*Une déclaration qualitative d'une intention qui doit être satisfaite par la solution implémentée. Elle est accompagnée d'au moins une justification et une mesure d'importance. TOGAF 9.1* [103]	p.79
Principe Généraux	*Un principe que l'approche satisfera indépendamment de l'entreprise ou du contexte.*	p.79
Principe Spécifique	*Un principe spécifique au contexte et/ou à la volonté des parties-prenantes de l'organisme mettant en place cette démarche.*	p.79
Processus disciplinaire	*Il s'agit d'un ensemble d'activités structurées (parfois appelées tâches), qui produisent un service dans un contexte spécifique. Les processus métier sont souvent représentés par l'utilisation de flowchart contenant des séquences d'activités entrelacées par des points de décision et un fork-join. Une séquence d'activités est un enchaînement d'activités orienté selon les transitions.*	p.45
Relation de structuration	*C'est une relation, i.e. une propriété des modèles qui relie plusieurs éléments entre eux. Elle a une sémantique et elle peut être de plusieurs types. Elle caractérise la structure du modèle qui reflète l'architecture du système étudié.*	p.50
Responsable de synchronisation	*Individu ou équipe en charge de mettre en place une solution concrète de synchronisation de modèles au sein de sa structure. Il joue le rôle de l'architecte système (cf. Chapitre I2.2. Missions de l'architecture système), cependant il se positionne au niveau de l'architecte d'entreprise* [49], [103].	p.40
Sémantique	*C'est une branche de la linguistique qui étudie les signifiés, ce dont on parle, ce que l'on veut énoncer. En informatique comme en linguistique, la sémantique désigne le lien entre un signifiant, le programme, et un signifié, objet mathématique qui dépendra des propriétés que l'on souhaite connaître du programme.*	p.16
Sponsor	*Individu ou équipe décisionnelle de l'entreprise qui formule une demande pour mettre en place une démarche collaborative entre des disciplines d'ingénierie. Cette entité décisionnelle doit financer les activités liées à sa demande.*	p. 40
Sûreté de fonctionnement	*Elle (dependability, SdF) consiste à évaluer les risques potentiels, prévoir l'occurrence des défaillances et tenter de minimiser les conséquences des situations catastrophiques lorsqu'elles se présentent.*	p.27
Sûreté de fonctionnement	*C'est l'ensemble des aptitudes d'un système à remplir une fonction requise au moment voulu, pendant la durée prévue, sans dommage pour lui-même et son environnement.*	p.27
Synchronisation de modèles	*C'est un cadre théorique permettant d'établir une correspondance entre les contenus de modèles. Elle part du constat que les modèles ne peuvent pas être comparés directement dans leur formalisme respectif du fait de leur hétérogénéité. Elle propose donc d'abstraire le contenu des modèles dans un formalisme commun et de comparer ces abstractions. L'objectif de cette comparaison est d'assurer la cohérence des modèles, ou tout au moins, de permettre aux différentes disciplines d'ingénierie de se mettre d'accord sur leurs désaccords.*	p.9, 42

Syntaxe	*La syntaxe est, à l'origine, la branche de la linguistique qui étudie la façon dont les mots se combinent pour former des phrases ou des énoncés dans une langue. En informatique, la syntaxe définit des règles d'agencement des lexèmes (en informatique, ce sont des entités lexicales d'un langage informatique) en des termes plus complexes, souvent des programmes. Ces règles permettent de définir des expressions bien formées et de décider du respect, ou du non-respect, de la grammaire formelle d'un langage.*	p.16
Système étudié	*Le système dont le cycle de vie est à l'étude dans notre contexte. ISO 15288* [35]	p.43
Système	*Combinaison d'éléments interagissant entre eux qui sont organisés pour atteindre un ou plusieurs objectifs. ISO 15288* [35]	p.43
Traçabilité	*Capacité à suivre l'historique, l'utilisation, ou la localisation d'un flux d'informations au moyen d'une identification enregistrée. La traçabilité est représentée par des traces.*	p.73
Trace	*Ensemble de liens de traçabilité concernant une même activité (ici, les activités sont l'abstraction, la comparaison et la concrétisation)*	p.73
Transformation de modèles	*C'est une des techniques clés de l'ingénierie des modèles. Elle consiste à prendre en entrée des modèles (source) et à fournir en sortie des modèles (cibles). Généralement un seul modèle source est utilisé et un seul modèle cible est fourni.*	p.69
Usecase	*Elément permettant de capturer les exigences des systèmes de manière informelle. Il décrit ce que les systèmes sont supposés faire. UML* [26]	p.79
Vérification des conditions de validation	*Etape de lancement d'une itération, elle vérifie si les conditions de validation sont respectées. Si une règle (règle 1, 2 et 3) n'est pas validée, elle en informera les ingénieurs et les architectes. Si la règle 4 n'est pas validée, l'itération pourra commencer et les ingénieurs seront informés de la relation de cohérence perdue. Si toutes les règles sont validées, alors la vérification autorisera l'exécution de la synchronisation de modèles.*	p.64
Vue d'architecture (View)	*Résultat exprimant l'architecture d'un système sous l'angle des préoccupations spécifiques du système ISO 42010 [3]. Une vue d'architecture est une représentation qui met en évidence une préoccupation (ou une partie d'une préoccupation) identifiée par les parties prenantes sur le système concerné.*	p.43

LISTE DES FIGURES ET DES TABLEAUX

TABLE DES ILLUSTRATIONS

LISTE DES TABLES

LISTES DES PROGRAMMES

ANNEXES

1. RESULTATS ET ANALYSE DE L'ENQUETE SUR LES PRATIQUES DE CONCEPTION D'ARCHITECTURE ET DE SURETE DE FONCTIONNEMENT

L'enquête a ciblé certaines questions sur les interactions entre les disciplines d'architecture système et de sûreté de fonctionnement. Les réponses ont été regroupées en trois catégories : au niveau humain, au niveau des outils et des langages et au niveau des pratiques de modélisations. De nombreux témoignages ont permis de narrer des situations quotidiennes et de mettre en avant des problèmes survenus lors d'échanges.

Au niveau humain, les disciplines, notamment transverses, coopèrent avec de nombreuses parties prenantes. Pour mieux percevoir le volume des interactions, les praticiens (de l'enquête) ont tenté d'énumérer les disciplines d'ingénierie interagissant avec la conception d'architecture système et la sûreté de fonctionnement, cf. Figure 173. 20 parties prenantes ont été citées en interaction avec l'architecture système et 17 en interaction avec la sûreté de fonctionnement, dont 11 communes. L'architecture système joue un rôle de médiation et de coopération avec les autres disciplines pour recueillir et fournir les résultats les plus pertinents possibles.

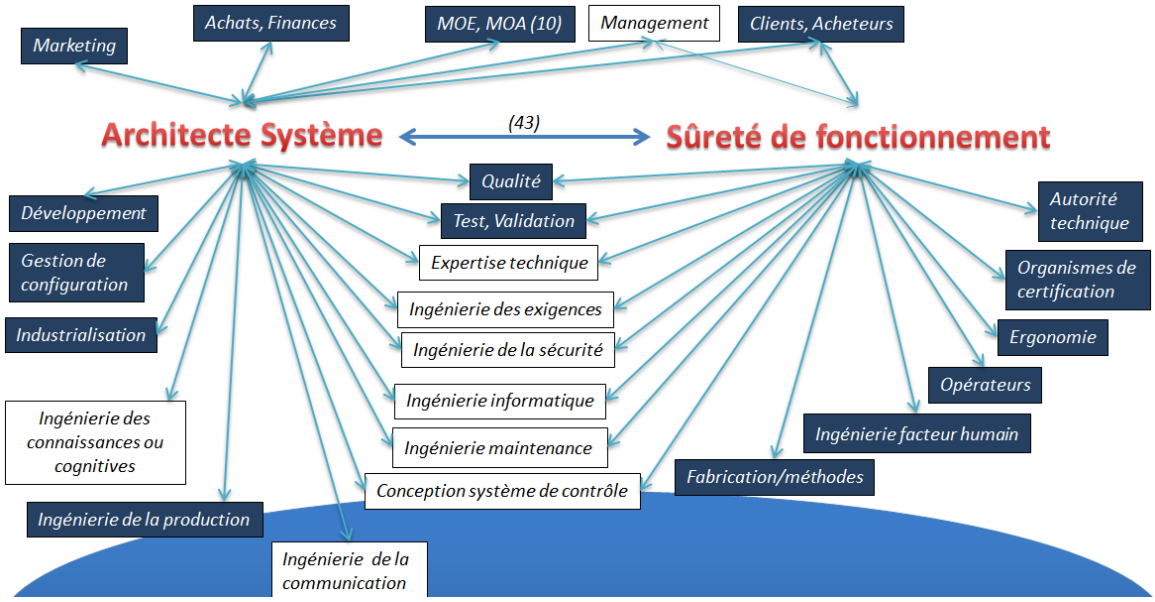


Figure 173 Interactions entre les disciplines d'ingénierie et les parties prenantes

L'enquête a montré l'importance de la communication entre ces deux disciplines :

- Les disciplines travaillent sur des aspects du système qui sont communs notamment les études fonctionnelles et physiques.
- Les disciplines doivent discuter et partager leurs points de vue sur le système.
- Ces deux disciplines sont transverses aux disciplines techniques. Elles ont intérêt à tenir un discours cohérent pour interagir avec les disciplines techniques.
- La communication des résultats en sûreté de fonctionnement est primordiale, ses analyses n'ont de sens que si elles sont partagées avec les autres disciplines.

- Les interactions permettent de valider les modèles, d'assurer une meilleure exhaustivité, apportent de la créativité et permettent le transfert d'information.
- Les interactions entre disciplines permettent également d'intervenir plus tôt dans les processus notamment pour la sûreté de fonctionnement et d'aller vers une gestion des versions des modèles améliorée.
- La communication permet à l'architecture de considérer les contraintes et les exigences de sûreté de fonctionnement au plus tôt dans les processus et à la sûreté de fonctionnement d'orienter ses études en tenant compte des contraintes de l'architecture du système.
- Une organisation des interactions peut permettre d'assurer que les disciplines étudient la même version des modèles du système.
- Une organisation des interactions peut permettre de construire et de mesurer la cohérence des études pour une meilleure efficacité et une meilleure complétude.

A noter qu'aucune personne ne souhaite un regroupement des études des deux disciplines. Ceci s'explique par la nécessité de séparer les préoccupations, les disciplines défendent différents intérêts sur le système qui sont parfois contradictoires. Leurs travaux ne peuvent pas être menés conjointement.

Concrètement, sur 100% des praticiens ayant répondu à l'enquête, les communications effectuées dans les projets sont réalisé par :

Des échanges documentaires	74%
Des réunions d'avant-projet	56%
Des réunions régulières tout au long du projet	68%
Des échanges et partages de modèles	17%

Les interactions entre ces disciplines, telles que menées aujourd'hui, posent de nombreux problèmes et risques pour le système en devenir. Les praticiens témoignent des problèmes auxquels ils sont confrontés :

Des conflits d'intérêt entre les disciplines	28%
Mauvaises interprétations des concepts, modèles et vocabulaire	25%
Incohérences entre les modèles représentant le même système	58%
Impact par la consommation importante de ressources	58%
Difficulté de maitrise de la complexité	49%
Mauvaise définition des périmètres d'études	38%

Extrait de situations problématiques encourus fournies par des répondants :

Témoignage n°1 : Par un architecte système, secteur spatial.
« Etude d'un système sur une configuration erronée car l'architecture système avance plus vite que les études de SDF. ».

Témoignage n°2 : Par un ingénieur de sûreté de fonctionnement dans un secteur peu standardisé.
« Par exemple, un concepteur - intégrateur de stations de compression de gaz qui sollicite notre entreprise (alors que sa conception est finalisée) parce qu'il y a une exigence "FMDS" dans son cahier des charges, le client ne sait pas trop ce que cela signifie et son client final lui a dit de consulter des sociétés spécialisées. Dans ce cas, nous réalisons les études souhaitées mais elles servent uniquement à compléter le dossier de cet industriel vis-à-vis de son client. Les données d'entrée (taux de défaillance composant par exemple) sont issues des bases de données publiques dont la représentativité reste faible. Sauf erreur grossière de la conception, nous arrivons à démontrer que les objectifs FMDS sont atteints et donc il n'y a pas de plus-value pour le concepteur lui-même.»

Témoignage n°3 : Par un ingénieur de sûreté de fonctionnement.
L'ingénieur de sûreté de fonctionnement a « l'obligation … de re-modéliser l'équipement (aux erreurs prêts, sans suivi des modifications) alors que ce travail a déjà été fait auparavant » dans d'autres outils. »

La suite de cette partie va s'intéresser aux langages et logiciels exploités pour mener les études et analyses des disciplines d'ingénierie. Les outils collaboratifs permettant l'interaction entre les disciplines sont étudiés au Chapitre I.

On a constaté que de nombreux éditeurs se sont positionnés sur des outils permettant l'accompagnement des activités de l'architecte système ou de l'ingénieur sûreté de fonctionnement, cf. Figure 174. Il y a eu une réelle explosion, ces 15 dernières années, d'éditeurs de logiciels. Aujourd'hui, ils sont très nombreux. Il existe une grande variété d'outils qui gagnent des parts de marché suivant le secteur d'activité dans lequel ils ont le mieux percé. Dans les deux disciplines, on peut remarquer qu'aucun outil émergeant significativement sur plusieurs secteurs d'activités.

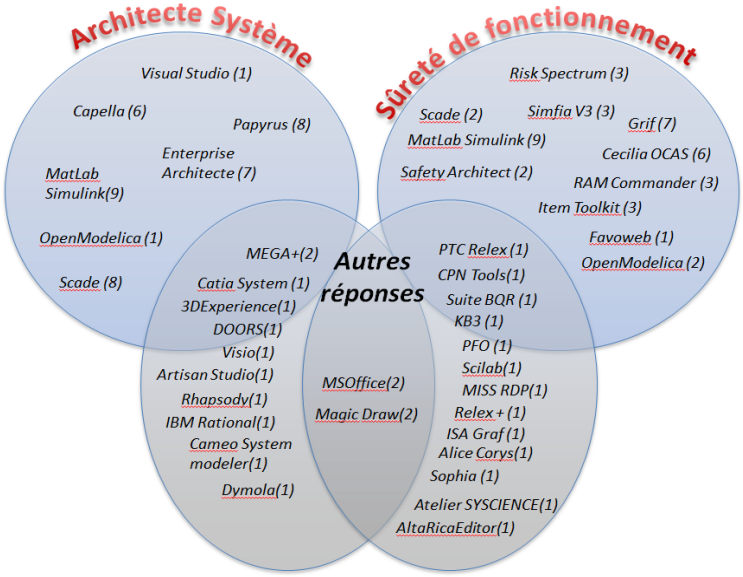


Figure 174 Résultat d'enquête d'usage des logiciels en architecture système et en sûreté de fonctionnement

Remarque : Tous les outils n'ont pas les mêmes fonctionnalités, leurs regroupements sont ceux des répondants et ne sont pas toujours justifiés.

Les langages de modélisation (ou les métamodèles) sont employés dans les logiciels cités pour permettre de construire des modèles spécifiques. Les langages décrivent comment les modèles doivent : être construits, se comporter, être évalués, être gérés, … Chaque langage a ses particularités, un objectif d'études particulières dans un contexte d'utilisation particulier.

On remarque une prédominance de certains langages parmi les outils utilisés. UML/SysML et les outils de traitement de texte sont les outils les plus employés par l'architecture système. Notons que UML/SysML sont des langages à base de modèle contrairement aux outils de traitement de texte qui ne sont absolument pas intégrés. En sûreté de fonctionnement, c'est AltaRica [70], [71] (décliné sous plusieurs versions) et les outils de traitement de texte (ou tableur) qui sont les plus utilisés. La Figure 175 présente les langages selon leur emploi dans chaque discipline. Les valeurs apposées à côté des langages représentent l'effectif des réponses pour chaque discipline.

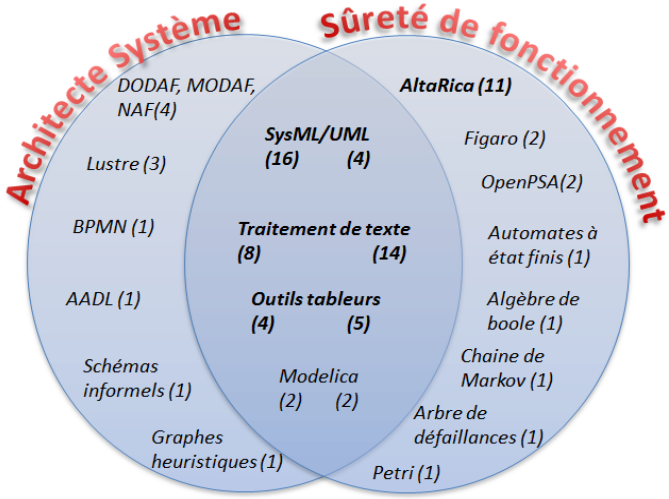


Figure 175 Les formalismes ou référentiels les plus employés Résultat d'enquête

L'enquête pose certaines questions concernant les interactions entre ces deux disciplines. Certaines idées émergent des réponses :

- *Il y a un besoin identifié d'interactions à double sens.*
- *Intervenir au plus tôt dans les* étapes *de développement des modèles de conception pour ouvrir des discussions plus en amont.*
- *Besoins d'assurer l'exhaustivité, la mise en cohérence entre les vues fonctionnelles et les vues dysfonctionnelles. Accès les études sur le produit réel, afin de garantir que les activités parlent toutes du même système.*
- *Besoins de boucles /interactions rapides pour permettre un gain de temps et d'efficacité. Ceci passe par le suivie des évolutions des modèles.*
- *Viser les mêmes objectifs à savoir trouver le système le plus adéquat en vue du besoin.*
- Parmi les réponses, des mots clés ont émergé comme *: l'ingénierie système, transversalité des disciplines, l'ingénierie collaborative, l'ingénierie concourante, certains parlent de modèle commun (pour les aspects réutilisabilité)*.

Les témoignages montrent certains problèmes. On peut constater que les interactions entre les disciplines ne sont pas optimales. Seuls certains chercheurs ont cité de nouvelles approches ou des buzzwords.

De nombreux répondants rapporttent qu'ils tentent physiquement d'interagir avec les autres disciplines (réunions, téléphone, email, visio, …).Cependant, aucun ne cite d'outil, de méthode, de standard, de langage dans leur référentiel métier pour soutenir leurs interactions et discussions.

2. APPLICATIONS DES POINTS DE SYNCHRONISATION ET MODELES DE COHERENCE

2.1. MODELE DE COHERENCE DU POINT DE SYNCHRONISATION 1

Le modèle de cohérence résultant de cette itération présente 3 relations de cohérence et 2 incohérences, il est le suivant :

$$R_{coh(ControlTrust,AS\&SdF)} = \{Re_1, Re_2\}\ avec\ Re_1 =< UC1A, 1 > et\ Re_2 =< UC1B, 1 >$$

$$R_{coh(Control\ Flight\ Path,AS\&SdF)} = \{Re_3, Re_4\}\ avec\ Re_3 =< UC2A, 1 > et\ Re_4 =< UC2B, 1 >$$

$$R_{coh(Determine\ Oritentattion,AS\&SdF)} = \{Re_5, Re_6\}\ avec\ Re_5 =< UC3A, 1 > et\ Re_6 =< UC3B, 1 >$$

$$R_{incoh(Determine\ Position\ and\ Heading,AS\&SdF)} = \{Re_7, Re_8\}$$
$$avec\ Re_7 =< UC4B, 1 > et\ Re_8 =< Inconnu, 0 >$$

$$R_{incoh(Engine\ Control,AS\&SdF)} = \{Re_9, Re_{10}\}$$
$$avec\ Re_9 = \{UC4A, 1\}\ et\ Re_{10} = \{UC5B, 0.25\}$$

2.2. MODELE DE COHERENCE DU POINT DE SYNCHRONISATION 2

Le modèle de cohérence résultant de cette itération contient 4 relations de cohérence et 1 incohérence, il est le suivant :

$$R_{coh(Lutter\ contre\ les\ incendies,AS\&SdF)} = \{Re_1, Re_2\}\ avec\ Re_1 =< AF1A, 1 > et\ Re_2 =< AF1B, 1 >$$

$$R_{coh(Détecter,AS\&SdF)} = \{Re_3, Re_4\}\ avec\ Re_3 =< AF2A, 1 > et\ Re_4 =< AF2B, 1 >$$

$$R_{coh(Lutter,AS\&SdF)} = \{Re_5, Re_6\}\ avec\ Re_5 =< AF3A, 1 > et\ Re_6 =< AF3B, 1 >$$

$$R_{coh(LuttertoDetecter,AS\&SdF)} = \{Re_7, Re_8\}$$
$$avec\ Re_7 =< Composition1A, 1 > et\ Re_8 =< Composition1B, 1 >$$

$$R_{Incoh(Lutter\ to\ Lutter,AS\&SdF)} = \{Re_9, Re_{10}\}$$
$$avec\ Re_9 =< Composition2A, 1 > et\ Re_{10} =< Inconnu, 0 >$$

2.3. MODELE DE COHERENCE DU POINT DE SYNCHRONISATION 3

Le modèle de cohérence résultant de cette itération contient 2 relations de cohérence et 2 incohérences, il est le suivant :

$$R_{coh(Fire\ Dection\ System\ to\ Detection\ System,AS\&SdF)} = \{Re_1, Re_2\}$$
$$avec\ Re_1 =< Composition1A, 1 > et\ Re_2 =< Composition1B, 1 >$$

$$R_{coh(Fire\ Dection\ System\ to\ Alert\ System,AS\&SdF)} = \{Re_3, Re_4\}$$
$$avec\ Re_3 =< Composition3A, 1 > et\ Re_4 =< Composition2B, 1 >$$

$$R_{Incoh(System\ to\ Dosposotif,AS\&SdF)} = \{Re_5, Re_6\}$$
$$avec\ Re_5 =< Composition2A, 1 > et\ Re_6 =< Inconnu, 0 >$$

$$R_{Incoh(System\ toReseau,AS\&SdF)} = \{Re_7, Re_8\}$$
$$avec\ Re_7 =< Composition4A, 1 > et\ Re_8 =< Inconnu, 0 >$$

2.4. APPLICATION D'UN POINT DE SYNCHRONISATION N°4

Avant d'appliquer la synchronisation, le Table 54 rappelle le contexte du point de synchronisation n°4.

Table 54 Point de synchronisation n°4 contextualisé

Disciplines d'ingénierie	**Architecture Système**	**Sûreté de fonctionnement**
Processus	ISO15288 [35]	ARP 4754 [91]
Activités	Architecture definition processes (Clause 6.4.4)	Aircraft FHA
Méthode	Conception interne du système	Modélisation architecture système
Vues	Diagramme interne de block : **Figure 96 Interactions des sous-systèmes dans un diagramme de blocks interne (cf. p. 126),** **Figure 97 Interactions des composants du sous-système Ensemble Capteur de feu (ECF) (cf. p. 127).**	Modèle AltaRica 3.0 : **Figure 115 Modélisation en S2ML du système (cf. p. 139).**
Eléments et Propriétés	Model, Block, Part, Port In/out, Connector, In/Out	Model, Block, Class, Assertion, *name*, *tabulation.*
Outil employés /Langages	Papyrus, SysML	AltaRica 3.0

La Figure 176, la Figure 177 et la Figure 178 synthétisent les étapes de la « méthode applicative de la synchronisation de modèles » (cf. Chapitre I5.2) sur le point de synchronisation 4. A des fins de synthèse, l'application ne concernera que le sous- système « Ensemble Capteur Feu » (aussi désigné « Detection system »).

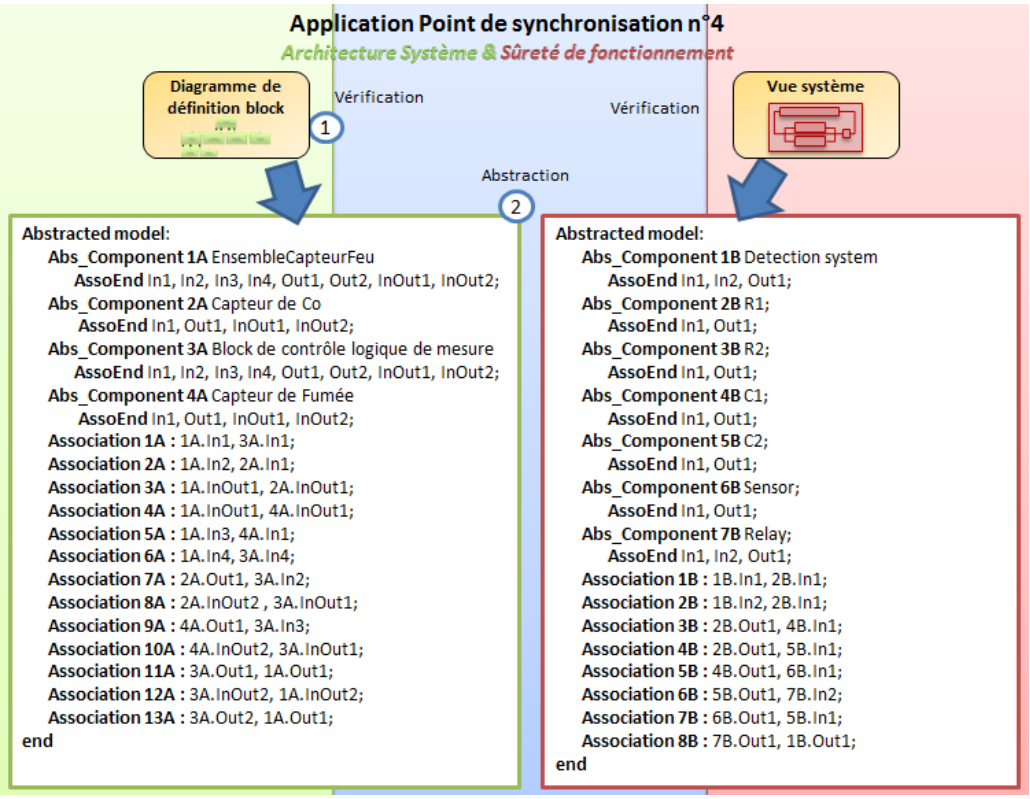


Figure 176 Application vérification et abstraction du point de synchronisation n°4

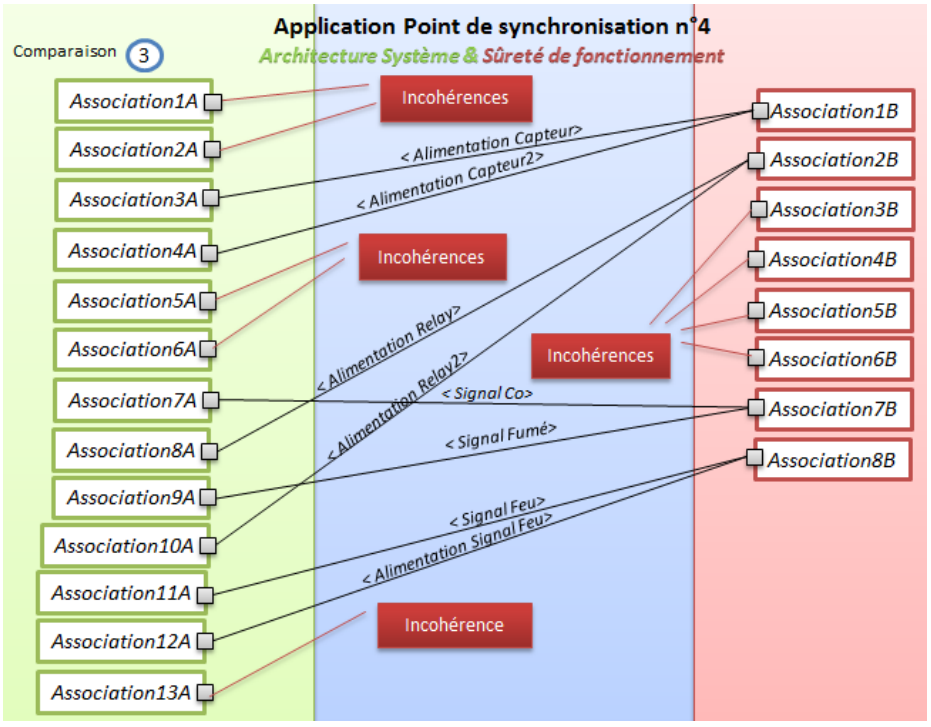


Figure 177 Application de la comparaison du point de synchronisation n°4

Neuf incohérences ont été identifiées concernant les relations d'association entre les composants physiques des deux modèles.

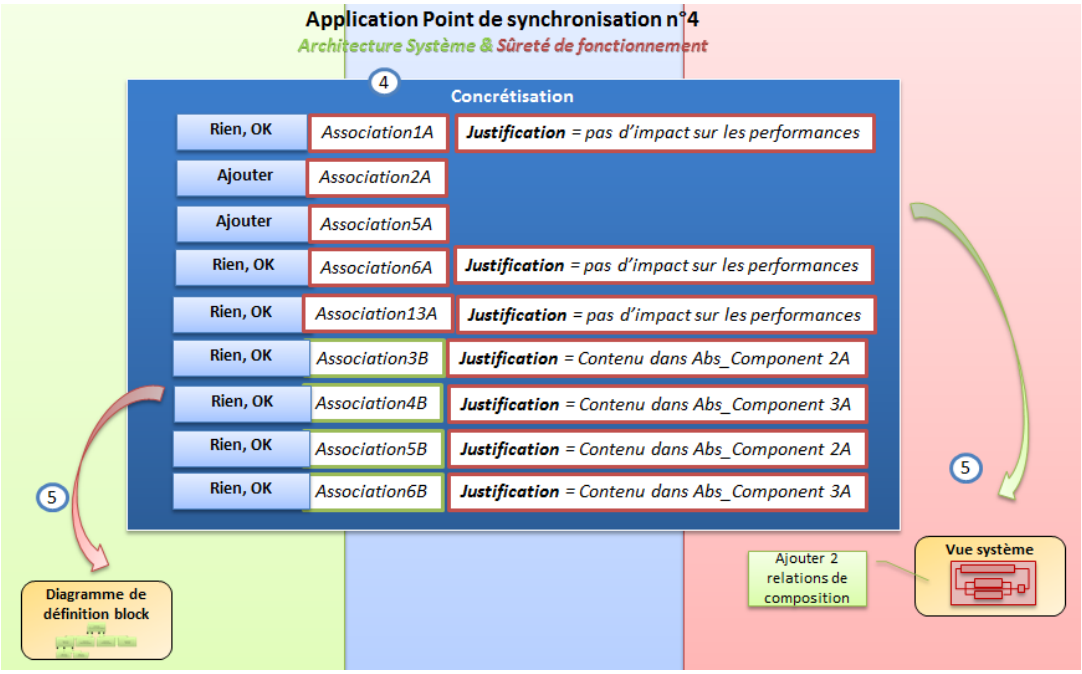


Figure 178 Application de la concrétisation du point de synchronisation n°4

Les éléments de traces des abstractions, de la comparaison et des concrétisations sont alors générés.

2.5. MODELE DE COHERENCE DU POINT DE SYNCHRONISATION 4

Le modèle de cohérence résultant de cette itération contient 8 relations de cohérence et 9 incohérences (dont 7 où l'écart entre les modèles est assumé), il est le suivant :

$$R_{coh(Alimentation\ Capteur,AS\&SdF)} = \{Re_1, Re_2\}$$
$$avec\ Re_1 =< Association3A, 1 > et\ Re_2 =< Association1B, 1 >$$

$$R_{coh(Alimentation\ Capteur2,AS\&SdF)} = \{Re_3, Re_4\}$$
$$avec\ Re_3 =< Association4A, 1 > et\ Re_4 =< Association1B, 1 >$$

$$R_{coh(Alimentation\ Relay,AS\&SdF)} = \{Re_5, Re_6\}$$
$$avec\ Re_5 =< Association8A, 1 > et\ Re_6 =< Association2B, 1 >$$

$$R_{coh(Alimentation\ Relay2,AS\&SdF)} = \{Re_7, Re_8\}$$
$$avec\ Re_7 =< Association10A, 1 > et\ Re_8 =< Association2B, 1 >$$

$$R_{coh(Signal\ Co,AS\&SdF)} = \{Re_9, Re_{10}\}$$
$$avec\ Re_9 =< Association7A, 1 > et\ Re_{10} =< Association7B, 1 >$$

$$R_{coh(Signal\ Fumé,AS\&SdF)} = \{Re_{11}, Re_{12}\}$$
$$avec\ Re_{11} =< Association9A, 1 > et\ Re_{12} =< Association7B, 1 >$$

$$R_{coh(Signal\ Feu,AS\&SdF)} = \{Re_{13}, Re_{14}\}$$
$$avec\ Re_{13} =< Association11A, 1 > et\ Re_{14} =< Association8B, 1 >$$

$$R_{coh(Alimentationn\ Feu,AS\&SdF)} = \{Re_{15}, Re_{16}\}$$
$$avec\ Re_{15} =< Association12A, 1 > et\ Re_{16} =< Association8B, 1 >$$

$$R_{Incoh(Etat\ de\ la\ zone,AS\&SdF)} = \{Re_{17}, Re_{18}\}$$
$$avec\ Re_{17} =< Association1A, 1 > et\ Re_{18} =< Inconnu, 0 >$$

$$R_{Incoh(Monoxyde\ de\ carbone,AS\&SdF)} = \{Re_{19}, Re_{20}\}$$
$$avec\ Re_{19} =< Association2A, 1 > et\ Re_{20} =< Inconnu, 0 >$$

$$R_{Incoh(Fumée,AS\&SdF)} = \{Re_{21}, Re_{22}\}$$
$$avec\ Re_{21} =< Association5A, 1 > et\ Re_{22} =< Inconnu, 0 >$$

$$R_{Incoh(Etat\ Lutte\ de\ la\ zone,AS\&SdF)} = \{Re_{23}, Re_{24}\}$$
$$avec\ Re_{23} =< Association6A, 1 > et\ Re_{24} =< Inconnu, 0 >$$

$$R_{Incoh(Etat\ de\ la\ zone\ surveillée,AS\&SdF)} = \{Re_{25}, Re_{26}\}$$
$$avec\ Re_{25} =< Association13A, 1 > et\ Re_{26} =< Inconnu, 0 >$$

$$R_{Incoh(SortieR1,AS\&SdF)} = \{Re_{27}, Re_{28}\}$$
$$avec\ Re_{27} =< Association3B, 1 > et\ Re_{28} =< Inconnu, 0 >$$

$$R_{Incoh(Sortie\ R2,AS\&SdF)} = \{Re_{29}, Re_{30}\}$$
$$avec\ Re_{29} =< Association4B, 1 > et\ Re_{30} =< Inconnu, 0 >$$

$$R_{Incoh(Sortie\ C1,AS\&SdF)} = \{Re_{31}, Re_{32}\}$$
$$avec\ Re_{31} =< Association5B, 1 > et\ Re_{32} =< Inconnu, 0 >$$

$$R_{Incoh(Sortie\ C2,AS\&SdF)} = \{Re_{33}, Re_{34}\}$$
$$avec\ Re_{33} =< Association6B, 1 > et\ Re_{34} =< Inconnu, 0 >$$

2.6. APPLICATION D'UN POINT DE SYNCHRONISATION N°5

Avant d'appliquer la synchronisation, le Table 55 rappelle le contexte du point de synchronisation n°5.

Table 55 Point de synchronisation n°5 contextualisé

Disciplines d'ingénierie	**Architecture Système**	**Sûreté de fonctionnement**
Processus	ISO 15288 [35]	ARP 4754 [91]
Activités	Architecture definition processes (Clause 6.4.4)	Aircraft FHA
Méthode	Décomposition du système	Modélisation de l'architecture système en vue d'analyse System CMA
Vues	Diagramme de définition block : **Table 20 Allocation des fonctions aux composants (cf. p. 126)**	Modèle AltaRica 3.0 : **Figure 110 Allocation des fonctions sur les composants physiques (cf. p. 133)**
Eléments et Propriétés	Model, Function, Association typed Composite, MemberEnd, Parent and Child.	Model, Block, Class, *name*, *tabulation.*
Outil employés /Langages	Papyrus, SysML	AltaRica 3.0

La Figure 179, la Figure 180 et la Figure 181 synthétisent les résultats des différentes étapes de la synchronisation sur une sous-partie de l'architecture du système. Elles concernent les relations d'héritage (ou d'allocation) entre les fonctions et les composants. Dans un premier temps, une étape de vérification est effectuée. Elle vérifie les traces liées aux relations de cohérence déjà identifiées sur les modèles fournis. Une seconde étape permet d'abstraire des deux vues, deux modèles sous un formalisme unique. Les éléments de type fonctionnel et composant ont été abstraits, puis les relations d'héritage ont été construites.

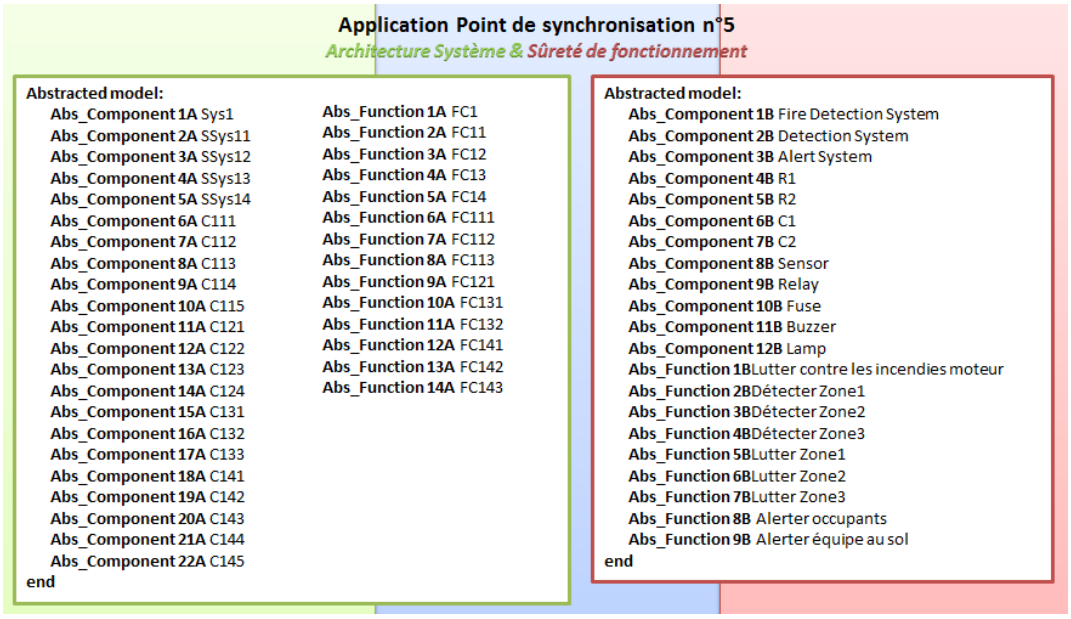


Figure 179 Application de l'abstraction des éléments du point de synchronisation n°5

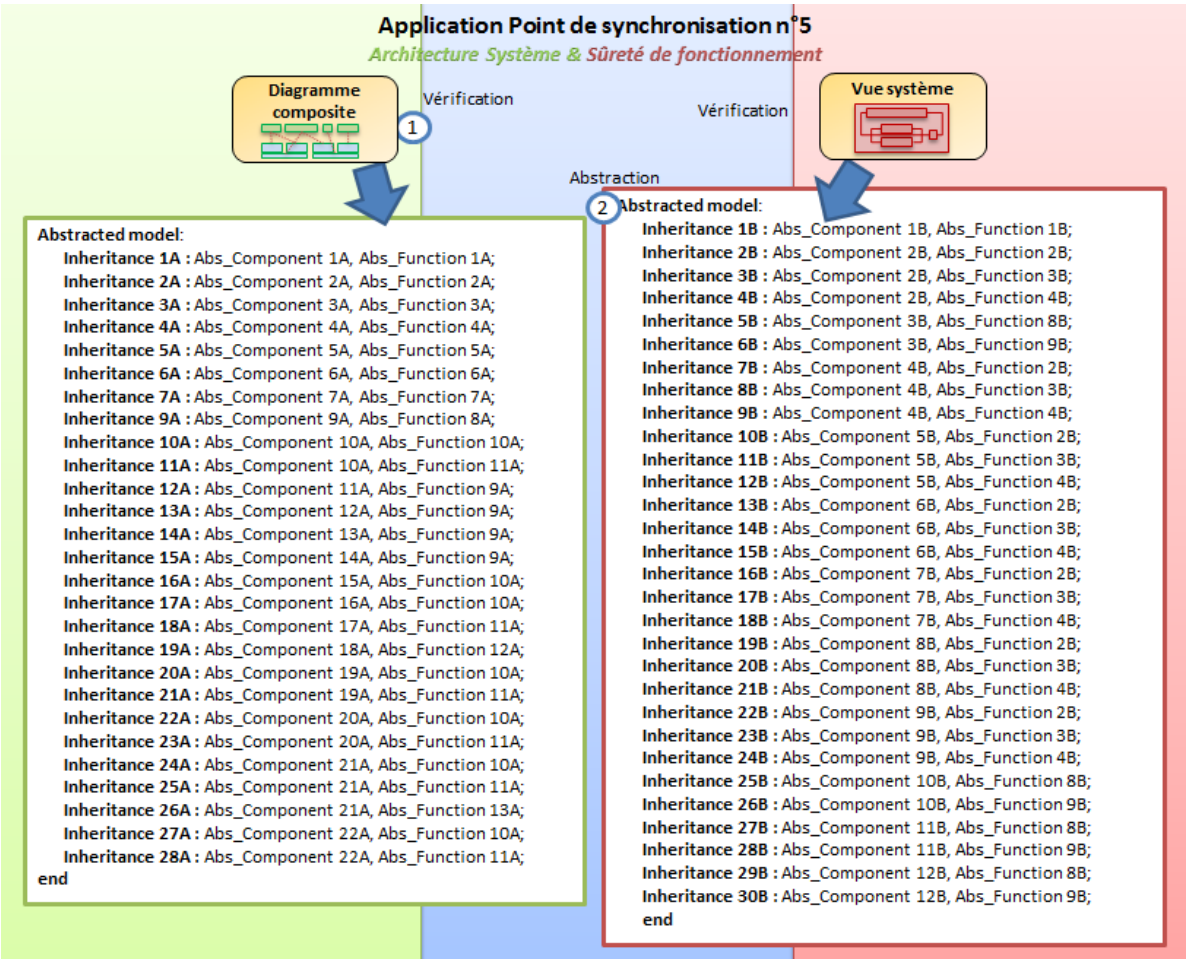


Figure 180 Application des abstractions du point de synchronisation n°5

Troisièmement, les ingénieurs des deux disciplines saisissent les relations de cohérence et lient les éléments de gauche à ceux de droite. A partir des règles de cohérence définies, des incohérences sont identifiées. Dans notre cas, 13 incohérences sont identifiées entre les relations d'héritage des deux disciplines.

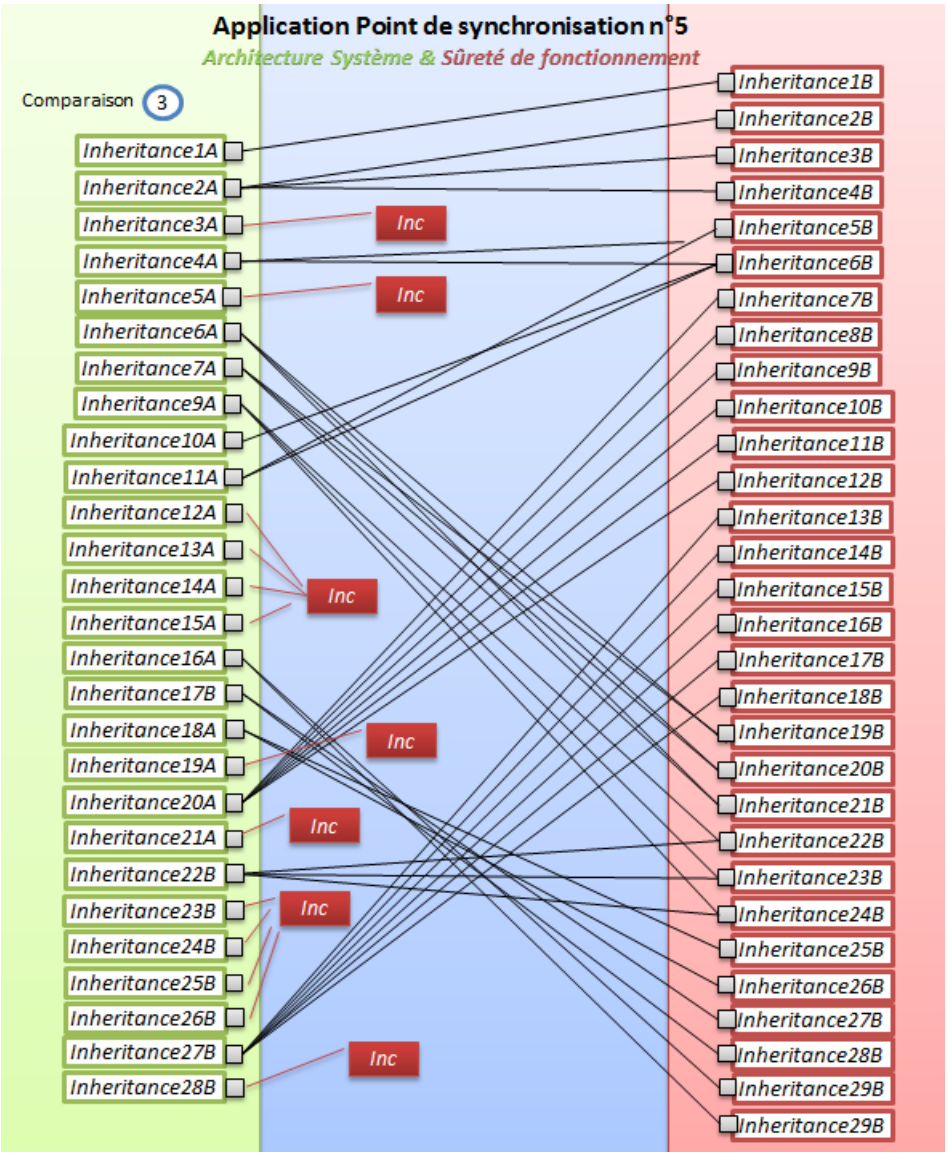


Figure 181 Application de la comparaison du point de synchronisation n°5

A partir de ces constats, une première étape de la concrétisation consiste à proposer des compromis pour résoudre ces incohérences, cf. Figure 182. Les ingénieurs discutent en réunion et proposent des compromis. Il s'agit d'ajouter, de supprimer une relation d'héritage ou d'accepter un écart entre les modèles.

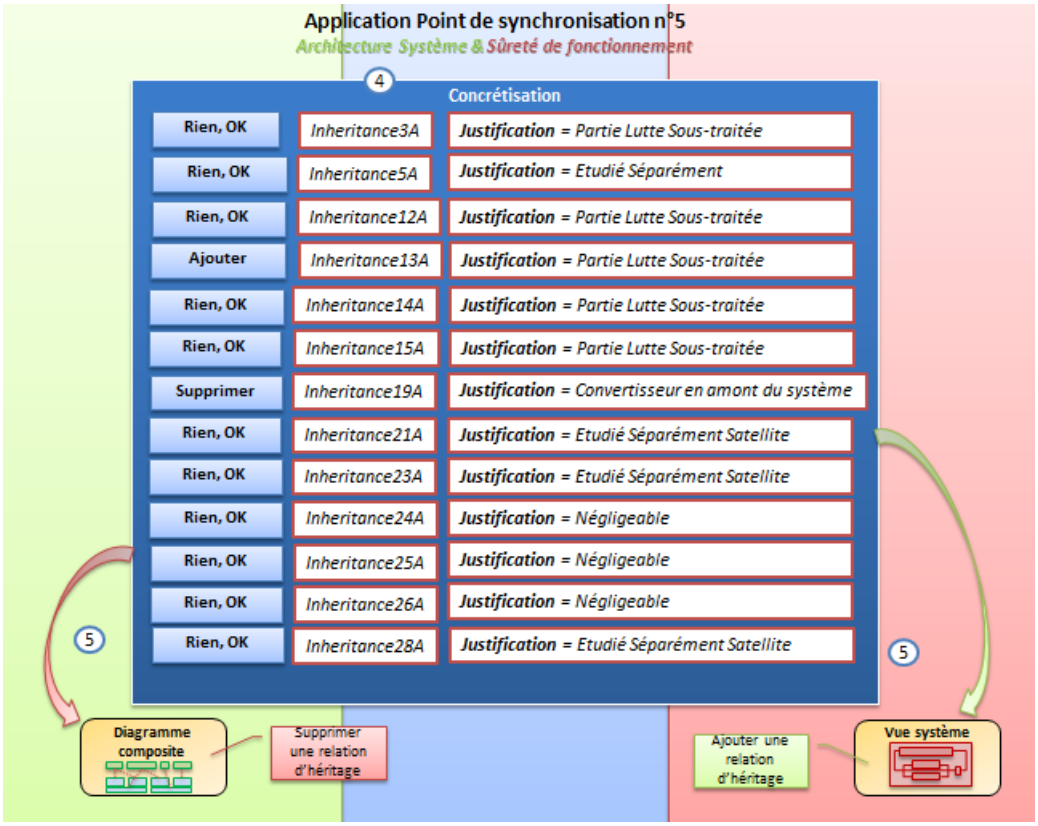


Figure 182 Application de la concrétisation du point de synchronisation n°5

Les éléments de traces des abstractions, de la comparaison et des concrétisations sont alors générés.

3. TRANSFORMATIONS DE MODELES IMPLEMENTEES

3.1. TRANSFORMATION JAVA SYSML VERS ALTARICA 3.0

Le Programme 3 présenté ci-dessous fait référence au Chapitre VI3

```
01              // generate an atomic node (a leaf node without sub-nodes)
02      public String genAtomicNode(Alt_Atomic_Node node) {
03              String res = "";
04              res = "class " + node.type + newline;
05
06              // Déclaration des états
08              res = res + "  //state declaration" + newline;
09              for(Alt_StateVariable sv : node.statesVariables) {
10                      res = res + "     Boolean " + sv.getName() + "(init = false); " +
newline;
11              }
12              // Déclaration les variables de flux In
13              res = res + "//flow declaration" + newline;
14              for(Alt_Port altPoIn : node.inputs) {
15                      res = res + "     Boolean " + altPoIn.getName() + "(reset = true);" +
newline;
16              }
17              // Déclaration les variables de flux Out
18              for(Alt_Port altPoOut : node.outputs) {
19                      res = res + "     Boolean " + altPoOut.getName() + "(reset = false);"
+ newline;
20              }
21              // Déclaration des évènements
22              res = res + "event " + newline;
23              for(Alt_StateVariable sv : node.statesVariables) {
24                      res = res + "     " + sv.getName() + "_occurs; " + newline;
25              }
26              // Définition des transitions
27              res = res + "transition " + newline;
28              for(Alt_StateVariable sv : node.statesVariables) {
29                      res = res + "     " + sv.getName() + "_occurs: " + sv.getName()+" ==
false -> " + sv.getName() + ":=true ; " + newline;
30              }
31              // Définition des assertions
32              res = res + "assertion " + newline + node.assertion + newline;
33              res = res + "end " + newline;
34              return res;
35      }
```

Programme 3 Extrait de la classe « Alt_Atomic_Node »

3.2. ABSTRACTION - TRANSFORMATION QVT-O D'UN DIAGRAMME DE USECASE VERS UNE LISTE HIERARCHIQUE

Le Programme 4 présenté ci-dessous fait référence au Chapitre VI4.1

```
modeltype uml uses "http://www.eclipse.org/uml2/5.0.0/UML";
modeltype absUsecasMM uses "http://www.absUsecaseMM.com";
transformation TranfoUC2FunctOp(in source: uml, out target: absUsecaseMM);

main() {//Find the root objects and invoke the mapping to transform them
        source.rootObjects()[Model].map modelToAbstractedModel();
}
```

```
mapping uml::Model::modelToAbstractedModel():absUsecaseMM::AbstractedList{
	//init
	init { log("Arrived in mapping transformation");}
	//population
		//call of UseCaseToAbs_UseCase transformation
	Abs_UseCase+= self.packagedElement[uml::Usecase].map UseCaseToAbs_UseCase();
	end { log("Leaving the mapping transformation");}
}
```

```
mapping uml::Usecase::UseCaseToAbs_UseCase():absUsecaseMM::Abs_UseCase {
	//population
	result.name:=self.name;
	};
}
```

Programme 4 Extrait du programme de transformation QVT-O UML vers une liste abstraite

3.3. ABSTRACTION - TRANSFORMATION QVT-O D'UN DIAGRAMME BDD VERS UNE LISTE HIERARCHIQUE

Le Programme 5 présenté ci-dessous fait référence au Chapitre VI4.2

```
modeltype sysml uses "http:\\www.papyrusuml.org/SysML/1" ;
modeltype absBlockMM uses "http://www.absUsecasMM.com";
modeltype uml uses "http://www.eclipse.org/uml2/5.0.0/UML";
transformation TranfoBlock2Component(in source: sysml, out target: absBlockMM);

main() {//Find the root objects and invoke the mapping to transform them
	source.rootObjects()[Model].map modelToAbstractedModel();
}
```

```
mapping sysml::Model::modelToAbstractedModel():absUsecasMM::AbstractedList{
	//init
	init { log("Arrived in mapping transformation");}
	//population
		//call of UseCaseToAbs_UseCase transformation
	Block += self.packagedElement[sysml::Block].map BlockToAbs_Component();
	end { log("Leaving the mapping transformation");}
}
```

```
mapping sysml::Block::BlockToAbs_Component():absUsecasMM::Abs_Component{
	//population
	result.name:=self.name;
	Block += self.packagedElement[sysml::Block].map BlockToAbs_Component();
	};
}
```

Programme 5 Extrait du programme de transformation QVT-O UML vers un modèle hiérarchisé

3.4. ABSTRACTION - TRANSFORMATION QVT-O D'UN DIAGRAMME IBD VERS UN MODELE DE CONNEXION

Le Programme 6 présenté ci-dessous fait référence au Chapitre VI4.3

```
modeltype sysml uses "http:\\www.papyrusuml.org/SysML/1" ;
modeltype absBlockConnectedMM uses "http://www.absUsecasMM.com";
modeltype uml uses "http://www.eclipse.org/uml2/5.0.0/UML";
transformation TranfoBlock2Component(in source: sysml, out target: absBlockMM);

main() {//Find the root objects and invoke the mapping to transform them
	source.rootObjects()[Model].map modelToAbstractedModel();
}
```

```
mapping sysml::Model::modelToAbstractedModel():absUsecasMM::AbstractedList{
	//init
	init { log("Arrived in mapping transformation");}
	//population
		//call of UseCaseToAbs_UseCase transformation
	Block += self.packagedElement[sysml::Block].map BlockToAbs_Component();
	Connector += self.packagedElement[sysml::Connector].map ConnectorToAssociation();
	end { log("Leaving the mapping transformation");}
}
```

```
mapping sysml::Block::BlockToAbs_Component():absUsecasMM::Abs_Component{
	//population
	result.name:=self.name;
	Block += self.packagedElement[sysml::Block].map BlockToAbs_Component();
	};
```

```
mapping sysml::Connector::MemberEndToAssociationEnd():absUsecasMM::Abs_Component{
	//population
	result.name:=self.name;
	AssociationEnd += self.packagedElement[sysml::Port].map MemberEndToAssociationEnd();
}
```

```
mapping sysml::Port::MemberEndToAssociationEnd():absUsecasMM::Abs_AssociationEnd{
	//population
	result.name:=self.name;
	};
}
```

Programme 6 Extrait du programme de transformation QVT-O UML vers un modèle d'association

3.5. ABSTRACTION - TRANSFORMATION PYTHON D'UN MODELE ALTARICA 3.0 VERS UNE LISTE HIERARCHIQUE

Le Programme 7 présenté ci-dessous fait référence au Chapitre VI4.4

```
# 1) Imported modules
import sys
import re
# 2) ComponentList
class ComponentList:
    # This class implements data structures to store AltaRica Block or Class
    # as well as some basic manipulation functions
    def __init__(self):
        self.content = []
        self.classcontent = []

    def GetComponent(self, order):
        return self.content[order - 1]

    def GetComponent(self, order):
        return self.content[order - 1]
    def GetClassComponent(self, order):
        return self.classcontent[order - 1]
    def SetComponent(self, value, kind):
        self.content.append(value)
        self.SetClassComponent(kind)

    def SetClassComponent(self, value):
        self.classcontent.append(value)

# 3) XMIWriter
class XMIWriter:
```

```
    # this class implements functions to write component into a Ecore file
    def ExportAtTxtFormat(self, componentList, fileName):
        try:
            output = open(fileName, "w")
        except:
            sys.stderr.write('Unable to open file "%s"\n' % fileName)
            sys.stderr.flush()
            return
        self.WriteAtEcoreFormat(componentList, output)
        output.close()
    def WriteAtEcoreFormat(self, componentList, output):
        output.write( "<?xml version=\"1.0\" encoding=\"UTF-8\"?>")
        output.write("\n")
        output.write("<mmsource:root xmi:version=\"2.0\"
xmlns:xmi=\"http://www.omg.org/XMI\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-
instance\" xmlns:mmsource=\"http://www.mmsource.fr\"
xsi:schemaLocation=\"http://www.mmsource.fr ../MModelSource/MMSource.ecore\">")
        output.write("\n")
        for i in range (1, len (componentList.content)+1) :
            output.write( "  <element
xsi:type=\"mmsource:"+componentList.GetClassComponent(i)+"\" classblock =\"" +
componentList.GetComponent(i) + "\"/>")
            output.write("\n")
        output.write("</mmsource:root>"+"\n")
        output.flush()

# 4) AltaRicaReader
class AltReader:
    # This class implements functions to read a AltaRica Component from a file
    def ImportAtTxtFormat(self, fileName):
        try:
            input = open(fileName, "r")
        except:
            sys.stderr.write('Unable to open file "%s"\n' % fileName)
            sys.stderr.flush()
            return
        componentList = self.ReadAtTxtFormat(input)
        input.close()
        return componentList
    def ReadAtTxtFormat(self, input):
        flag = 0
        row = -1
        componentList = ComponentList()
        for line in input:
            if flag == 1:
                line = line.strip(' \t\n\r;')
                words = re.findall(r"[\w']+", line)
                for word in words:
                    if not(word in componentList.classcontent):
                        componentList.SetComponent(word,words[0])
            flag=0
            if line.find("block ") == 0:
                flag=1
        return componentList

# 5) Main
reader = AltReader()
writer = XMIWriter()
componentList = reader.ImportAtTxtFormat("source.alt")
if componentList is None:
    sys.exit()
writer.ExportAtTxtFormat(componentList,"target.xmi")
```

Programme 7 Prototype de transformation de modèles AltaRica 3.0 vers des listes de composant en Ecore

3.6. CONCRETISATION - TRANSFORMATION QVT-O D'UN MODELE ABSTRAIT ET SES COMPROMIS VERS UN MODELE DE CLASSE UML

Le Programme 8 présenté ci-dessous fait référence au Chapitre VI5

```
modeltype uml "strict" uses "http://www.eclipse.org/uml2/3.0.0/UML";
modeltype ecore "strict" uses "http://www.eclipse.org/emf/2002/Ecore";

transformation TransformationConcretisation( in modelSource: uml, in sourceComment: uml,
out target: uml);
main() {
	modelSource.rootObjects()[Model] -> map Model2Model();
}
```

```
mapping uml::Model::Model2Model() : uml::Model {
	// init
	init { log("Arrived in mapping transformation for concretisation"); }

	if(IsOwnerofComment(self)){
		//int
		log("--Arrived in mapping commentary");
		result.ownedComment :=
getownedComments(self.ownedElement[uml::NamedElement]);
		// end
		log("--Leaving the mapping commentary");
	};
	// population
	result.name := self.name + " - Updated";
	result.packagedElement := self.packagedElement[uml::Class]->sortedBy(name).map
Class2Class();//call and create new mapping if other NamedElement (as Class) should be
annotated by comment
	result.ownedType := self.ownedType[uml::Association]->sortedBy(name).map
Asso2Asso();
	// end
	end { log("Leaving the mapping transformation"); }
}
```

```
mapping uml::Class::Class2Class() : uml::Class{
	result.name := self.name;
	result.ownedAttribute := self.ownedAttribute;
	if(IsOwnerofComment(self)){
		result.ownedComment :=
getownedComments(self.ownedElement[uml::NamedElement]);
	}endif;
	result.nestedClassifier := self.nestedClassifier[uml::Class]->sortedBy(name).map
Class2Class();
	result.nestedClassifier:= self.nestedClassifier[uml::Association]-
>sortedBy(name).map Asso2Asso();
	result.generalization:= self.generalization->sortedBy(name).map Gen2Gen();
}
```

```
mapping uml::Association::Asso2Asso() : uml::Association{
	result.name := self.name;
	if(IsOwnerofComment(self)){
		result.ownedComment :=
getownedComments(self.ownedElement[uml::NamedElement]);
	}endif;
}
```

```
mapping uml::Generalization::Gen2Gen() : uml::Generalization{
		result.ownedComment := getownedComments(self.ownedElement[uml::NamedElement]);
}
```

Programme 8 Extrait de la transformation « Concrétisation de commentaire sur un modèle UML »

4. EVALUATION DES TRAVAUX SUR LA SYNCHRONISATION DE MODELES

Cette annexe a pour objet de présenter quelques éléments d'évaluation des travaux de thèse présentés en vue de :

- Vérifier la pertinence des travaux par rapport aux besoins industriels ;
- Présenter les travaux connexes autour des contributions ;
- Quantifier les contributions.

4.1. ADEQUATION DES TRAVAUX AVEC LE BESOIN

La synchronisation de modèles a été formalisée et construite pour répondre aux problèmes d'interactions multidisciplinaires. Ces problèmes ont été introduits au Chapitre I1.4. Dans le manuscrit l'hypothèse est faite que ces problèmes sont réels et impactent significativement les projets industriels de systèmes complexes. Afin de valider cette hypothèse et de vérifier que le besoin est réel, plusieurs actions ont été menées :

- Des contacts directs avec des industriels ;
- Une enquête sur les pratiques de l'architecture système et de la sûreté de fonctionnement ;
- Une étude de l'orientation de la DGA ;

Une synthèse rapide des retours de ces actions est présentée dans les sections qui suivent.

4.1.1. DISCUSSIONS AVEC LES INDUSTRIELS

De nombreux échanges et discussions ont été menés avec des industriels, des groupes de travail, etc. Les présentations des industriels témoignent de pratiques et des problèmes rencontrés. Très souvent, ils confirment rapidement les problèmes traités dans cette thèse. Les industriels constatent des problèmes d'interactions au niveau des processus, des pratiques et au niveau humain. Ils recherchent des solutions méthodologiques outillées prêtes à l'emploi.

Les entreprises rencontrées individuellement sont : Safran Tech, Safran Landing System, Thales R&T, DGA.

De nombreux autres industriels ont été rencontrés dans les groupes de travail ou journées thématiques :

- CT SV2S (Sûreté, Vérification et Validation Système et Système de systèmes) de l'AFIS :
  - Airbus Group, Naval Group, PSA, Thales, Safran Tech, IRT SystemX, etc.
- RM (Recherche Méthodologique) de l'IMdR :
  - Satodev, EDF R&D, Total, Université Nancy, etc.
- SFSF (Séminaire Francilien de la Sûreté de Fonctionnement) organisé par CentraleSupelec et l'ENS CACHAN.
- Journée « jeunes chercheurs et jeunes ingénieurs » de l'IMDR.

De tous les échanges avec les entreprises liées à la conception de systèmes complexes, tous ont confirmé la problématique justifiant nos travaux. Les entreprises rencontrées s'identifient et semblent intéressées par les solutions proposées.

4.1.2. ENQUETE SUR LES PRATIQUES DES DISCIPLINES D'INGENIERIE ET LEURS INTERACTIONS

Une enquête sur les pratiques industrielles a été effectuée avec le soutien du comité technique SV2S (Sûreté, Validation, Soutien des Systèmes) de l'AFIS, l'Association Française d'Ingénierie Système conjointement et du comité technique du LambdaMu20, l'IMdR, l'Institut de la Maîtrise de Risques. Les résultats sont présentés en Annexe 1. Les témoignages confirment la problématique avérée concernant les rapports de conflit entre les disciplines d'ingénierie. Cette section caractérise uniquement la population interrogée.

L'enquête avait deux objectifs :

- Capitaliser des informations sur les pratiques des architectes systèmes (ou ingénieur système) et des ingénieurs sûreté de fonctionnement.
- Identifier les pratiques industrielles et les problèmes induits par des interactions entre les deux disciplines d'ingénierie.

L'enquête a capitalisé 49 retours fournissant ainsi de nombreux témoignages, cf. Figure 183. Chaque retour répondait à une cinquantaine de questions qui abordaient les thématiques d'expertise, de processus, de modélisation. Parmi la population interrogée, 33 témoignent de leur expérience en sûreté de fonctionnement et 37 en architecture système. 22 praticiens ont transmis des retours dans les deux disciplines.

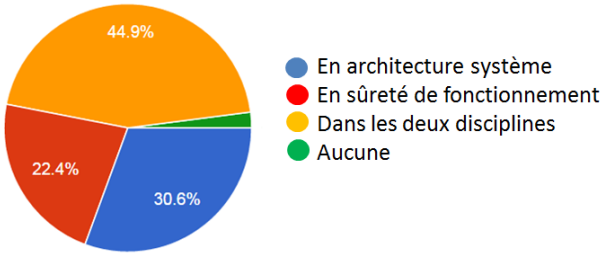


Figure 183 Répartition des répondants selon leurs expériences dans les disciplines

Au total 34 entreprises ont apporté leurs opinions aux questions posées :

Airbus Defence and Space, Alstom, ANSYS, AREVA TA, Assystem, CEA DRT, Centre de Recherche en Automatique de Nancy (CRAN), CS SI, Dassault Aviation, Dassault Systèmes, DCNS, DGA TA, DGA TT, DNVGL, EDF R&D, EISTI, ENSTA ParisTech, European Aviation Safety Agency, FMDS industrie, Giat Industries - Nexter Systems, Ksdf conseil, LGM, Magneti Marelli Systèmes Electroniques, MBDA, ONERA, RATP, Rolls-Royce Civil Nuclear, SAFRAN NACELLES, Samares Engineering, Schneider Electric, SECTOR, SYSCIENCE, Thales, Zodiac Aerospace.

La répartition de l'expérience des praticiens en ingénierie système dans l'une des deux disciplines est présentée dans la Figure 184.

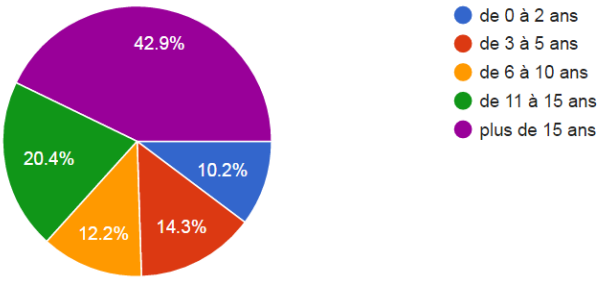


Figure 184 Années d'expériences des répondants en conception d'architecture ou/et en sûreté de fonctionnement

Ce résultat conforte la qualité des réponses, car près de 43 % des répondants ont plus de 15 ans d'expérience en entreprises.

L'enquête était dynamique, c'est-à-dire que selon les réponses du répondant, les suivantes étaient différentes. Au total, 7 scénarios étaient possibles :

- Scénario 1 : Le praticien est expérimenté en architecture système et n'utilise pas de modèle pour mener ses études.
- Scénario 2 : Le praticien est expérimenté en architecture système et utilise des modèles pour mener ses études.
- Scénario 3 : Le praticien est expérimenté en sûreté de fonctionnement et n'utilise pas de modèle pour mener ses études.
- Scénario 4 : Le praticien est expérimenté en sûreté de fonctionnement et utilise des modèles pour mener ses études.
- Scénario 5 : Le praticien est expérimenté en architecture système et en sûreté de fonctionnement et n'utilise pas de modèle pour mener ses études.
- Scénario 6 : Le praticien est expérimenté en architecture système et en sûreté de fonctionnement et utilise des modèles pour mener ses études.
- Scénario 7 : La personne n'ayant aucune expérience en architecture système et en sûreté de fonctionnement.

La Figure 185 présente la répartition de la population interrogée selon leurs profils.

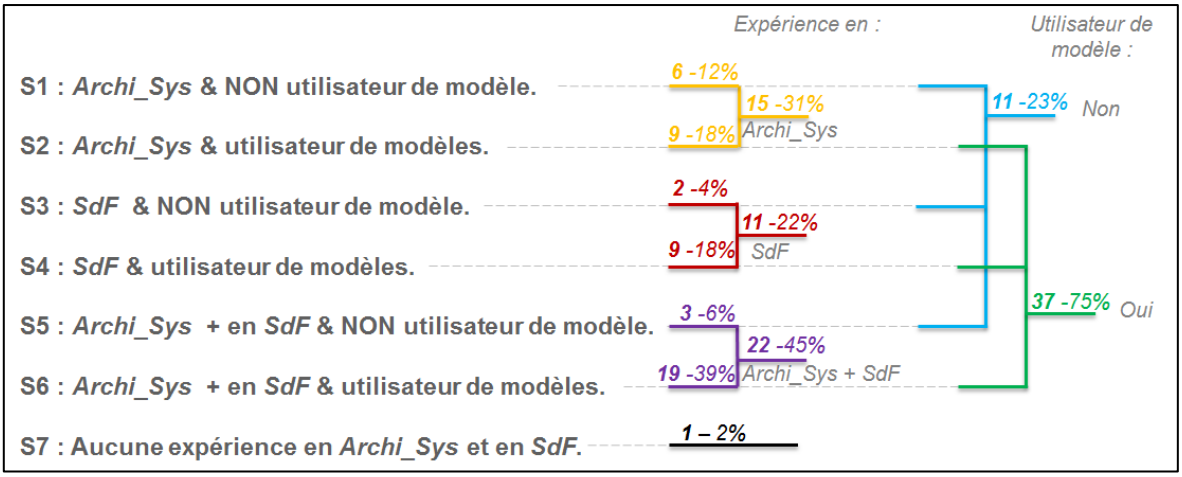


Figure 185 Répartition des répondants selon leurs profils

Selon le profil, entre 9 et 50 questions ont été posées aux praticiens. Les questions étaient majoritairement semi-ouvertes (questions à choix multiples avec la possibilité de rédiger une réponse « autres »).

Ces travaux ont apportés des résultats fructueux. Des discussions avec le CT SV2S à l'AFIS ont permis de faire ressortir une nouvelle perspective : *« Effectuer une nouvelle fois l'enquête sur une population plus grande (en Europe, à l'international) et catégoriser par secteurs d'activité ».*

L'enquête a apporté d'importants retours sur des pratiques industrielles en matière d'architecture système et de sûreté de fonctionnement. Elle a également permis d'apprécier le niveau de connaissance et le niveau de confiance des ingénieurs et des architectes dans leurs pratiques. Des contrastes importants ont été observés entre les entreprises et entre les secteurs d'activités.

4.1.3. ECHANGE SUR L'ORIENTATION DE S&T A LA DGA

La direction générale de l'armement, co-finançant cette thèse, produit tous les deux ans un document sur les orientations et les priorités de recherche de la DGA en sciences et sur les technologies[38].

Concernant les technologies transverses, i.e. « les études visant à améliorer les méthodes et outils nécessaires à la maîtrise des systèmes complexes et des systèmes de systèmes », est un enjeu de la DGA pour :

- « Garantir la performance et la disponibilité des constituants »,
- Garantir « des méthodes et outils permettant de spécifier, concevoir »,
- « Porter à maturité les technologies transverses émergentes ».

Elle met notamment une priorité sur les activités et les disciplines : « **Analyse et comparaison d'architecture système** », « Sûreté de fonctionnement des systèmes informatiques embarqués », « Résilience des systèmes de systèmes », …

Une thématique technologique prioritaire également émise par la DGA en Ingénierie de l'information et Robotique (I2R) est : « **Des systèmes sûrs, fiables et robustes.** » Elle souhaite mettre l'accent sur « Sécurité informatique, intégrité et authentification des données et des interactions, supervision des réseaux, réseaux ad-hoc, sûreté de fonctionnement des logiciels, systèmes hybrides et embarqués… ».

Comme le témoignent les experts de DGA TT[39], les équipes d'ingénierie système ont pour mission de mettre en place une équipe d'ingénierie système transverse aux activités de spécification, suivi de conception et vérification et validation des programmes à la DGA pour les futurs systèmes d'armes et de défense au profit des états-majors français.

Un groupe de travail « sûreté de fonctionnement » interne à la DGA aborde notamment comment lier efficacement les travaux de l'architecte système et de l'ingénieur de sûreté de fonctionnement, pour chacune des étapes d'une opération d'armement.

Comme le témoigne un expert en ingénierie système de la DGA, la démarche mise en œuvre par l'équipe sûreté de fonctionnement à la DGA TA[40] vise à évaluer et éventuellement à comparer les performances de sûreté de fonctionnement présentées par les industriels fournisseurs de la DGA. La première étape consiste, à partir des données techniques issues de la documentation industrielle et des données techniques fournies par l'architecte DGA du système, à réaliser une modélisation dysfonctionnelle du système. Cette étape est d'autant plus facile quand les données d'entrée sont complètes. Une formalisation fournie sous forme de modèles d'architecture et peu ouvertes à des interprétations divergentes entre architectes et ingénieurs de sûreté de fonctionnement simplifie le travail de modélisation dysfonctionnelle. La modélisation dysfonctionnelle pratiquée à DGA TA couvre les points de vue fonctionnels, organiques (vue physique des éléments du système) et zonaux (localisation géographique).

Le principal enjeu pour DGA TA est de diminuer le temps consacré à la construction du modèle : instaurer un dialogue le plus en amont possible entre l'architecte et l'ingénieur SdF pour valider les hypothèses de haut-niveau sur les futurs modèles, privilégier les opérations graphiques sur l'écriture de code, utiliser la transformation de modèles pour récupérer le

[38] [web38] DGA (2014). "Document de Présentation de l'Orientation de la S&T Période 2014 2019", sur le site Ministère des armées, Bagneux. Consulté le 10 août 2017. https://www.ixarm.com/IMG/pdf/post_dga_2014_2019.pdf

[39] DGA TT : Direction générale de l'Armement Technique Terrestre

[40] DGA TA : Direction générale de l'Armement Technique Aéronautique

maximum d'information de la démarche Ingénierie Système, en restant conscient des limites actuelles de ces transformations : la modélisation dysfonctionnelle nécessite de décrire des niveaux bas d'interaction entre composants non traités par la modélisation d'architecture ; comprendre comment le système fonctionne ou comment le système dysfonctionne amène à construire des points de vue d'architecture différents et donc une logique d'exploitation différente des modèles.

Les thématiques de recherche et les équipes spécialistes à la DGA ciblent de nombreux problèmes scientifiques notamment la mise en cohérence et la comparaison de modèles entre les architectures systèmes et les modèles de sûreté de fonctionnement. C'est notamment une des raisons qui a motivé la DGA à cofinancer ces travaux.

4.2. EVALUATION DES PRINCIPALES CONTRIBUTIONS

Cette section tente de quantifier le travail effectué. Elle caractérise les principaux résultats présentés dans cette thèse : la conceptualisation, la méthodologie, les cas d'application et des travaux d'implémentation.

Pour chaque contribution, différents critères sont choisis pour dénombrer les concepts utilisés ou introduits, les méthodes proposées, les démonstrations effectuées etc.

4.2.1. TRAVAUX DE CONCEPTUALISATION

Les concepts et les formalismes étudiés et proposés font référence au cadre conceptuel présenté au Chapitre II. Ils sont classifiés ci-dessous.

Quarante-quatre concepts ont été présentés. Ils sont tous rattachés à une définition et utilisés par un métamodèle ou un formalisme. Ils sont classés dans la Table 58.

Table 56 Quantification des concepts utilisés dans le cadre conceptuel de synchronisation de modèles

Cadre conceptuel		**Définition des disciplines d'ingénierie**		**Définition des concepts d'architecture**	
Concept repris	*Concepts définis*	*Concepts repris*	*Concepts définis*	*Concept repris*	*Concepts définis*
1	3	8	4	0	4
Configuration des interactions multidisciplinaire		**Mise en cohérence**		**Application de synchronisation**	
Concepts repris	*Concepts définis*	*Concept repris*	*Concept définis*	*Concept repris*	*Concepts définis*
1	3	0	1	1	5
Traçabilité et histoires des modèles			**Méthodologie de synchronisation de modèles**		
Concepts repris		*Concept définis*	*Concepts repris*		*Concepts définis*
3		0	4		6

Ces concepts sont rattachés à plus de 28 métamodèles ou formalismes (cf. Table 57) définis pour appliquer la synchronisation de modèles à plusieurs niveaux : processus, point de vue, abstraction, comparaison, concrétisation, traçabilité et mise en cohérence, concept pivot d'architecture, etc.

Table 57 Mécanismes ou points de vue proposés dans le mémoire selon les niveaux conceptuels

Niveau	**Mécanismes ou points de vue**
Processus	Définition des disciplines d'ingénierie, Cas d'utilisation de la synchronisation, Définition et ordonnancement des besoins de synchronisation, Ordonnancement des points de synchronisation, Application de la synchronisation de modèle
Point de vue	Définition des points de synchronisation, Principes d'interactions

Niveau	Mécanismes ou points de vue
Abstraction	Mapping, Ordonnancement des fonctions de transformation, Transformations de modèles.
Comparaison	Algorithme de comparaison.
Concrétisation	Mapping, Ordonnancement des fonctions de transformation Transformations de modèles.
Traçabilité et mise en cohérence	Modèle de cohérence et tous les modèles de traces sur la synchronisation.

Un effort important a été fourni pour la définition des concepts et leur formalisation. Cependant, des éclaircissements peuvent être apportés sur certains concepts et métamodèles, notamment la construction de la traçabilité, des modèles de cohérence, la comparaison et la formalisation des concepts d'architecture et des métamodèles pivots.

4.2.2. METHODOLOGIE DE SYNCHRONISATION

L'évaluation de la méthodologie et ses étapes font référence au Chapitre III. Elle est présentée ci-dessous.

La méthodologie de synchronisation de modèles propose une manière de configurer puis d'appliquer des cas de synchronisation de modèles portant sur l'architecture du système. Elle utilise tous les concepts et formalismes de la partie conceptualisation. Elle permet d'utiliser les principes d'abstraction, de comparaison et de concrétisation des modèles. Elle produit des modèles de cohérence tout au long des processus.

La méthodologie est découpée en 5 étapes successives contenant des sous-étapes, tel que présenté dans la Table 58.

Table 58 Liste des étapes et des méthodes proposées par la méthodologie

Etape de la méthodologie	***Sous étape / méthodes***
Principes d'interactions multidisciplinaires	- Choix des disciplines d'ingénierie ; - Définition opérationnelle du projet de synchronisation ; - Etat actuel des interactions et des problèmes observés ; - Définition de l'état cible ; - Evaluation de l'adéquation du besoin avec l'approche ; - Etudes complémentaire.
Définition des contextes d'ingénierie	- Définition des contextes d'ingénierie - Définition des besoins de synchronisation
Configuration de la synchronisation	- Formalisation des points de synchronisation
Application de la synchronisation	- Application de la synchronisation de modèles
Suivi des cohérences	- Construction des traces - Interprétations des traces

Bien que la méthodologie tente d'être exhaustive, certaines étapes ne sont pas abordées, notamment la définition des concepts pivot d'architecture et le lien avec les métamodèles pivots.

Les étapes proposées par la méthodologie sont très générales, il est parfois nécessaire ou judicieux de les adapter au besoin particulier d'une entreprise ou de les raffiner pour plus d'efficacité. La méthodologie autorise cela, mais aucun cas n'a été présenté.

Toutes les étapes de la méthodologie ont été illustrées sur plusieurs exemples simples. Cependant, ces exemples sont ponctuels et ne montrent pas le déroulement de la méthodologie. C'est pourquoi, une application complète de la méthodologie a été effectuée sur un cas d'étude.

4.2.3. APPLICATION SUR CAS D'ETUDE

L'évaluation de l'application de la méthodologie sur le cas d'étude fait référence aux Chapitre IV et Chapitre V.

Le cas d'étude est représentatif des études effectuées en entreprise car il provient d'études réelles. De plus, il a été comparé aux études menées par des industriels sur un système de détection incendie d'un hélicoptère de l'armée française, actuellement en service.

Le système a été entièrement conçu puis évalué sous des aspects de sûreté de fonctionnement durant la thèse. L'application utilise les termes et les référentiels utilisés dans le secteur aéronautique puisque le système est embarqué dans un hélicoptère. Les 5 étapes de la méthodologie ont été appliquées. Trois besoins de synchronisation ont d'abord été identifiés. A partir de cela, 5 points de synchronisation ont configurés puis appliqués sur des modèles du cas d'étude. Finalement, 27 incohérences et 57 relations de cohérence ont été établies. Parmi les 27 incohérences, aucune n'avait été identifiée avant de synchroniser les modèles ou les points de vue. La synchronisation apporte une forte valeur ajoutée dans les projets.

L'application présentée a cependant une faiblesse, elle n'illustre pas le déroulement des itérations successives exécutées sur un même point de synchronisation. Cela a peu d'impact sur l'application mais peut porter à confusion.

4.2.4. TRAVAUX D'IMPLEMENTATION

L'évaluation de l'implémentation fait référence au Chapitre VI. Les travaux d'implémentation ont permis de montrer la possibilité de construire des points de vue, un profil, des transformations de modèles pour l'abstraction et la concrétisation et des algorithmes de comparaison. Ils mettent en avant la faisabilité d'outiller la méthodologie de synchronisation de modèles.

Pour envisager un déroulement complet et outillé des étapes de la méthodologie ou simplement de la synchronisation, des efforts conséquents sont nécessaires, notamment pour l'intégration de l'ensemble des modèles, des formalismes et des algorithmes. Par ailleurs, plusieurs outils existants peuvent être employés pour effectuer certaines étapes de la méthodologie.

Les algorithmes de comparaison n'ont pas été implémentés. Ils ont tous été exécutés manuellement. Il existe pourtant des outils de comparaison de modèles cependant l'investigation n'as pas permis de faire ressortir une comparaison au niveau d'abstraction de nos modèles.

université PARIS-SACLAY

ÉCOLE DOCTORALE
INTERFACES
Approches Interdisciplinaires : Fondements, Applications et Innovation

Titre : Ingénierie système et Sûreté de fonctionnement: Méthodologie de synchronisation des modèles d'architecture et d'analyse de risques

Mots clés : Synchronisation de modèles, Cohérence, Ingénierie système basée sur les modèles, Evaluation de la sûreté de fonctionnement basée sur les modèles, Approche collaborative.

Résumé : L'organisation classique en silos disciplinaires des industries atteint ses limites pour maîtriser la complexité. Les problèmes sont découverts trop tard et le manque de communication entre les experts empêche l'émergence précoce de solutions. C'est pourquoi, il est urgent de fournir de nouvelles approches collaboratives et des moyens d' interactions entre les disciplines d'ingénierie, au début et tout au long du cycle de développement. Dans ce contexte, nous avons étudié l'approche synchronisation de modèles entre deux domaines d'ingénierie : la conception d'architecture de systèmes et la sûreté de fonctionnement. Elle a pour but de construire et maintenir la cohérence entre les modèles.

Ces travaux proposent, étudient et analysent une démarche collaborative de synchronisation de modèles. Ils tiennent compte des contextes d'études, des processus, des méthodes appliqués et des points de vue produits par les ingénieurs. Les contributions répondent à des problématiques au niveau des pratiques, des concepts, de la mise en œuvre, des applications et l'implémentation de la synchronisation de modèles.

Title : System engineering and dependability: methodology synchronization of models

Keywords : Models synchronization, Consistency, Model based system engineering, Model based safety assessment, Collaborative approach.

Abstract : Classical organization in disciplinary silos in the industry reaches its limits to manage and control complexity. Problems are discovered too late and the lack of communication between experts prevents the early emergence of solutions. This is why it is urgent to provide new collaborative approaches and ways to exchange the models contents between various engineering fields, early and all along the development cycle. In this context, we are particularly interested in a synchronization approach of models between two engineering fields: system architecture design and dependability analysis.

This work proposes a collaborative approach of synchronization of models. It takes into account the study contexts, applied processes, applied methods and viewpoint produced by engineers. Contributions address issues at levels of practices, concepts, implementation, applications and implementation of model synchronization.