Approche pour la déclinaison des exigences de sûreté des systèmes complexes.

sROMARIC GUILLERM[1], HAMID DEMMOU[1], NABIL SADOU[2]

[1] LAAS CNRS - Université de Toulouse ; UPS, INSA, INP, ISAE; LAAS
7 avenue du Colonel Roche, 31077 Toulouse, France
prenom.nom@laas.fr

[2] Supelec/IETR
Avenue de la Boulais, 35511 Cesson-Sevigne, France
prenom.nom@adresse.fr

Résumé – Cet article s'intéresse à la définition et à la gestion des exigences de sûreté des systèmes complexes. On constate que d'une part, l'ingénierie des exigences est l'un des processus les plus critiques de la conception système et d'autre part, que les propriétés de sûreté sont des propriétés émergentes qui résultent d'interdépendances entre les composants d'un système et de l'interaction avec son environnement. Il est donc nécessaire d'accorder une attention particulière aux exigences de sûreté. Ces propriétés doivent être considérées globalement au niveau du système complet si l'on souhaite qu'elles soient respectées. Elles ne peuvent être attribuées localement. L'approche proposée dans cet article permet de définir la propriété de sûreté au niveau système suite à des analyses de risque et ensuite décliner ces dernières au niveau des sous-systèmes. Elle est basée sur l'analyse AMDEC et les arbres de défaillance et formalisée grâce à l'utilisation d'UML.

Abstract **-** This paper deals with safety requirements definition and management of complex system. One of critical system engineering processes is requirements engineering. Safety is defined as a non functional requirement and is related to emergent system properties. These non-functional properties cannot be attributed to single system components. Indeed, they emerge as a result of integrating system components. So safety requirements must be formulated in the large (system level) and then declined in the small (sub-system level). The method proposed in this paper allows defining system safety requirements following a risk and hazard analysis and then declining them in sub-systems. It is based on Failure Mode, Effects, and Criticality Analysis and Fault Tree.

Mots clés – Ingénierie système, Ingénierie des exigences, Sûreté de fonctionnement, AMDEC, Arbres de Défaillances.
Keywords – System engineering Requirements engineering, Dependability, FMECA, Fault tree.

1 INTRODUCTION

Les systèmes actuels sont de plus en plus complexes [Chavalarias et al, 2008]. Ils sont capables de fournir des fonctions de haut niveau, dont le comportement global est difficile à prévoir et dont la structure présente un graphe d'interaction non-trivial, souvent pourvu de boucles de rétroactions, et associant la plupart du temps plusieurs technologies de par l'implication d'un grand nombre de constituants [Magee et de Weck, 2004].

Les processus de conception et d'évaluation de ces systèmes deviennent eux aussi de plus en plus complexes [Komi-Sirvio et Tihinen, 2003] en particulier les analyses de sûreté et de fiabilité [Avizienis et al., 2004].

L'analyse de plusieurs accidents (Ariane 5, Mars Polar Lander) a montré que les composants n'étaient pas eux mêmes défaillants en termes de non-satisfaction des exigences pour lesquelles ils ont été conçus. Les composants ont fonctionné exactement comme cela était prévu. Les problèmes proviennent des effets imprévus ou mal compris des comportements des composants sur le système dans son ensemble. Ce sont des erreurs dans la conception du système plutôt que dans la conception des composants (y compris l'analyse de la sûreté de ces composants). Notamment, il s'agit d'erreurs dans l'attribution et la traçabilité des fonctions globales du système au niveau des composants individuels.

Ceci démontre la nécessité d'une approche globale pour la conception système et plus particulièrement pour les analyses de sûreté de fonctionnement. Cette dernière doit être prise en compte d'une manière globale et non localement. En effet les propriétés de sûreté sont des propriétés émergentes qui résultent d'interdépendances existant dans le système et dans l'interaction avec son environnement. Il est absolument nécessaire que ces propriétés soient étudiées globalement au niveau du système complet si l'on souhaite qu'elles soient respectées. Il est alors nécessaire d'éliciter [Goguen et Linde, 1993] d'une manière globale les exigences de sûreté. Ces dernières seront ensuite déclinées, au fur et à mesure, au niveau local et doivent être satisfaites par les différents composants du système. Notons qu'il est essentiel de garantir une traçabilité [Gotel. et Finkelstein, 1994], [Sahraoui, 2005] dans la déclinaison des exigences globales à un niveau local.

De surcroit, ces propriétés importantes du système doivent être considérées dès le début de la conception. En effet, elles ne peuvent pas être introduites ou seulement mesurées a posteriori, et doivent être traitées le plus tôt possible pour limiter leurs impacts sur les délais et les coûts de conception. De plus elles ne peuvent pas être attribuées à des composants.

Un des processus les plus importants de l'ingénierie système est celui de l'ingénierie des exigences [Sommerville, 2006]. C'est une part très importante de l'ingénierie système qui est en charge de toutes les activités liées aux exigences : définition, traçabilité, modification, gestion en terme de maturité, etc.

Une exigence correspond à une expression de besoin bien formulée émanant du client ou de toutes autres parties prenantes en lien avec le système à développer. Elle transmet un besoin en fonctionnalité (exigence fonctionnelle) ou en qualité (exigence non-fonctionnelle) que doit satisfaire le produit qui est en train d'être conçu. Les exigences de sûreté sont considérées comme des exigences non fonctionnelles.

L'ingénierie des exigences comportent deux activités. La première concerne le développement avec l'élicitation, la documentation, l'analyse et la validation des exigences. La seconde s'intéresse à leur gestion avec la maintenance, la gestion des changements et la traçabilité des exigences. Les deux activités sont importantes pour la réussite d'un projet de conception.

L'approche présentée dans cet article est une partie d'un travail plus complet qui concerne l'intégration de la sûreté de fonctionnement dans les processus d'ingénierie système [Guillerm et al., 2009]. Le travail a permis de proposer une approche qui permet de prendre en compte les risques liés à l'intégration de plusieurs technologies. Les exigences de sûreté de fonctionnement sont définies non seulement localement mais globalement (niveau système). Cela revient à formuler ces exigences au niveau du système complet et, ensuite, à les décliner à des niveaux plus bas. L'intégration de la sûreté de fonctionnement dans les processus de l'ingénierie système offre un cadre structurant pour mener les analyses dans un contexte plus large que celui traditionnellement rencontré dans le milieu fiabiliste. Le travail devrait aboutir dans un cours terme à l'élaboration d'un guide méthodologique destiné aux ingénieurs système et ingénieurs « safety » pour une optimisation de la prise en compte de la sûreté de fonctionnement des systèmes complexes.

L'approche proposée dans cet article concerne les deux activités de l'ingénierie des exigences. L'article montre comment décliner les exigences de sûreté système en exigences de sûreté au niveau des composants (dérivation des exigences locales à partir des exigences système). L'approche combine des études AMDEC (Analyse des Modes de Défaillance de leur Effets et leur Criticité) [Buzzatto, 1999] et des analyses par Arbre de Défaillance (AdD) [Lee et al., 1985].

L'article contient 5 parties. La seconde présente le cadre de travail de notre méthodologie de gestion de la sûreté des systèmes complexes. L'approche de déclinaison des exigences est ensuite présentée dans la partie 3. Un exemple d'application est donné dans la quatrième partie. La dernière partie conclut le travail.

2 CADRE DE TRAVAIL

Deux principales approches de conception ont fait l'objet de nombreux travaux et ont été définies. Le cycle en V et ses variantes, [Forsberg & al., 1995], [Boehm, 1994], et l'approche processus.

L'approche processus est basée sur le fait que quelque soit la stratégie utilisée pour le développement d'un système, les activités restent les mêmes. Les processus techniques représentant ces différentes activités d'ingénierie système sont groupés en deux catégories principales : les processus de définition du système et les processus de vérification et de validation du système. Ils sont définis dans des normes d'ingénierie système (EEE-1220, EIA-632, ISO-15288, 2008).

L'approche processus est adoptée dans notre travail car elle est plus flexible que l'approche de développent en V. En effet, la vision processus ne contraint pas la séquence des activités de développement, contrairement aux approches basées sur des cycles [Meinadier, 2002]. Cette différence est la motivation pour le choix de l'approche processus, d'autant plus qu'il s'agit de systèmes complexes.

2.1 Ingénierie système

L'ingénierie système fournit un ensemble de concepts facilitant la conception de nouveaux systèmes. Il s'agit d'un processus collaboratif et interdisciplinaire de résolution de problèmes, supportant les connaissances, les méthodes et les techniques résultants des sciences et de l'expérience. L'ingénierie système est en fait un cadre de travail qui aide à définir un nouveau système satisfaisant l'ensemble des besoins des parties prenantes et étant acceptable pour l'environnement, tout en respectant la balance économique globale de la solution pour tous les aspects du problème et pour toutes les phases de développement et de vie du système. L'ingénierie système convient tout particulièrement aux problèmes complexes [Sahraoui et al., 2004].

2.2 La norme EIA-632

Un standard, couramment utilisé dans l'industrie ou le secteur militaire, est l'*EIA-632* [Spitzer, 2007]. Ce standard couvre le cycle de vie du produit depuis la capture des besoins jusqu'au transfert du système à l'utilisateur.

Le standard EIA-632 définie une approche d'ingénierie ou de réingénierie des systèmes, incorporant les bonnes pratiques constatées dans la seconde partie du XXème siècle. Il fournit une méthodologie d'ingénierie système à travers 13 processus regroupés en 5 catégories : *management technique*, *acquisition et fourniture*, *conception système*, *réalisation du produit* et *évaluation technique*. Un ou plusieurs sous-processus sont définis pour chacun de ces 13 processus et le développeur doit décider lesquels des 33 sous-processus appliquer.

Ce standard fournit un cadre de travail à notre étude, sur lequel s'appuie l'approche de prise en compte de la sûreté au sein de processus généraux d'ingénierie système.

Le standard EIA-632 définit la notion de produit final et produits contributeur. Les produits finaux sont une part d'un système qui réalise des fonctions opérationnelles et qui est livrée à l'acquéreur. Les produits contributeurs sont des éléments qui fournissent les moyens de mettre en service un produit final, de le maintenir en service ou le retirer.

La définition du système est donnée de manière hiérarchique définie par la notion de «building block».

2.3 Le concept de Building Block

La norme EIA-632 adopte une approche originale pour la décomposition du système. Le système final est décomposé en une hiérarchie de sous-systèmes définis en tant que modules. Le développement des modules de niveau inférieur est lancé dès que le module de niveau supérieur est complètement spécifié. Chaque module est alors considéré comme un produit qui a des caractéristiques et des exigences identifiées et fait l'objet d'un développement de même nature que le système global (Figure 1).

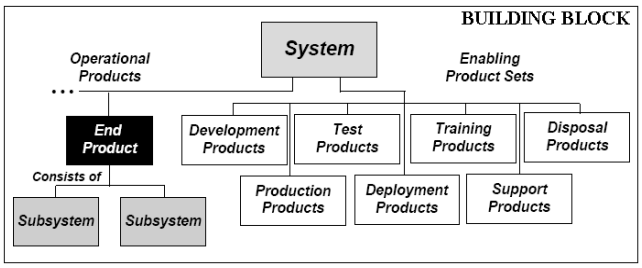

Figure 1. Concept de Building Block

Ainsi la solution définie à un block de niveau supérieur décrite par un certain nombre d'exigences constitue des exigences d'entrée pour le niveau inférieur (figure 2).
La décomposition se poursuit jusqu'à l'identification d'une des trois catégories de produits finaux :

- Produits finaux sur étagère,
- Produits finaux pouvant être implémentés directement,
- Produits finaux pouvant être fournis par un sous-traitant.

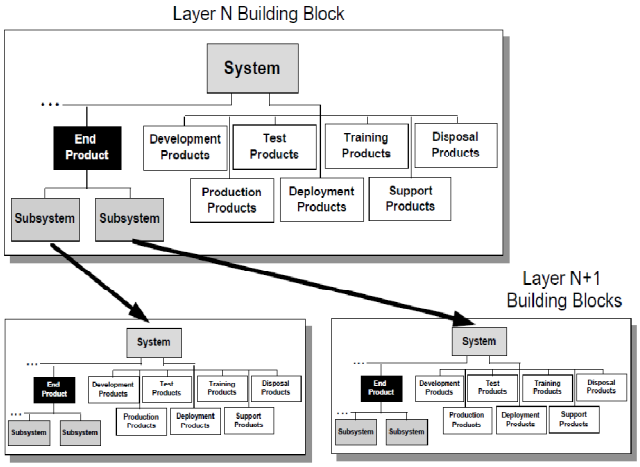

Figure 2. Building block multi-niveaux

3 DECLINAISON D'EXIGENCES DE SURETE DE FONCTIONNEMENT

Avant de présenter la méthodologie de déclinaison d'exigences de sûreté de fonctionnement, il convient de justifier le besoin d'une telle méthode, ainsi que de situer cette déclinaison parmi l'ensemble des processus d'ingénierie système.

3.1 Pourquoi cette déclinaison ?

La complexification des systèmes impose une évolution des études des propriétés de sûreté, afin d'assurer et de permettre les niveaux de confiance nécessaires. Pour une considération effective de la sûreté de fonctionnement par le processus de conception, il est impératif de considérer cette sûreté à travers des études globales dans les processus d'ingénierie système.
Nous pouvons rappeler les exemples de catastrophes qui prouvent à posteriori un manquement dans ce domaine, tel que l'explosion d'Ariane 5 ou le crash du Mars Polar Lander.
Une fois que les propriétés de sûreté ont été identifiées globalement (c'est-à-dire élicitées), celles-ci doivent être déclinées localement pour être effectivement réalisées par le système. Les propriétés locales associées aux sous-systèmes ou aux composants doivent être établies afin d'assurer les propriétés globales, ce qui implique un travail de traçabilité et des activités d'ingénierie des exigences.

3.2 Déclinaison vis-à-vis des processus d'ingénierie système

Nous avons noté dans l'introduction l'importance des processus d'ingénierie des exigences dans l'ingénierie système. L'ingénierie des exigences est considérée comme un processus crucial dans la conception des systèmes complexes.
Les exigences de sûreté de fonctionnement peuvent être classifiées comme des exigences non-fonctionnelles et sont liées aux propriétés émergentes du système. Définies au niveau du système, elles ne peuvent pas être attribuées à un seul constituant du système. Bien entendu, ces exigences non-fonctionnelles sont fondamentales pour le succès d'un projet de conception.

La méthodologie de déclinaison proposée permet de prendre en compte les exigences de sûreté de fonctionnement en facilitant leur traçabilité. En fait, elle concerne les deux processus principaux de la gestion des exigences : à la fois les activités de développement et celles de management. D'une part, la méthode aide à l'identification (c'est-à-dire l'élicitation) des exigences système de sûreté de fonctionnement. Puis, à l'aide de différentes analyses, elle permet de décliner les exigences globales de sûreté de fonctionnement en exigences locales, élicitées à leur tour. D'autre part, la traçabilité engendrée par la déclinaison répond parfaitement à un besoin des activités de management des exigences.

3.3 Positionnement par rapport à l'EIA-632

Pour bien situer la position de la méthodologie de déclinaison des exigences de sûreté de fonctionnement, la figure 3 reprend les principaux processus de l'EIA-632 qui entrent en jeu.

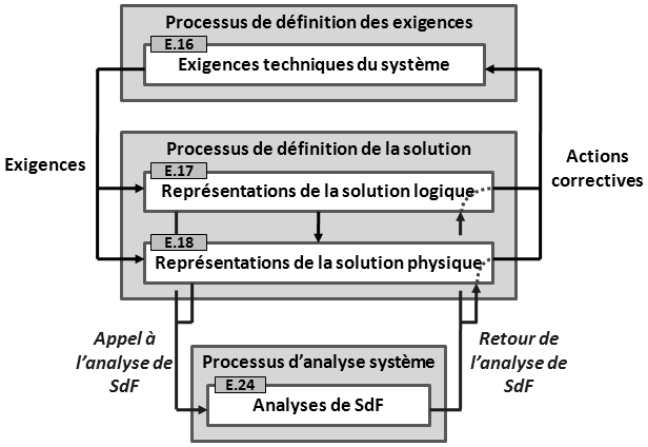

Figure 3. Evolution parallèle des processus de Définition des Exigences et de Définition de l'architecture

Au niveau des exigences de sûreté il existe 3 types de ressources [Romaric et al,. 2010]. Celles si peuvent provenir :

1. directement de l'acquéreur ou d'une partie prenante (autre que norme ou certification),
2. de norme à respecter ou de certification à satisfaire,
3. ou encore d'analyse de sûreté de fonctionnement.

Dans cet article on s'intéresse plutôt aux exigences du 3ème type. C'est-à-dire les exigences provenant d'analyse de risque et de sûreté de fonctionnement.

4 PRESENTATION DE LA METHODOLOGIE

Dans cette section, nous allons présenter la méthodologie de déclinaison d'exigences de sûreté de fonctionnement proposée. Nous rappelons brièvement les deux méthodes de sûreté de fonctionnement utilisées, qui sont l'AMDEC et l'analyse par arbres de défaillances.

4.1 Aperçu de la méthodologie

L'objectif de la méthodologie est de pouvoir lier les exigences de sûreté de fonctionnement définies au niveau du système avec celles définies au niveau des sous-systèmes. Ces liens correspondent alors à la déclinaison des exigences.
La méthodologie se décompose en 4 étapes, organisées et définies de la manière suivante :

- Etape 1 : analyse des défaillances au niveau du système complet qui peuvent conduire à une situation

catastrophique ou, tout du moins, à un événement non-souhaité.

- Etape 2 : analyses des causes des ces défaillances en s'appuyant sur l'architecture du système. Il s'agit de trouver les origines des défaillances système au niveau des sous-systèmes.
- Etape 3 : analyses des défaillances au niveau des sous-systèmes.
- Etape 4 : synthétiser la déclinaison des exigences de sûreté de fonctionnement à l'aide des informations des précédentes étapes.

La figure 4 résume le procédé de la méthode en montrant la séquence des différentes étapes et en spécifiant les informations d'entrées/sorties.

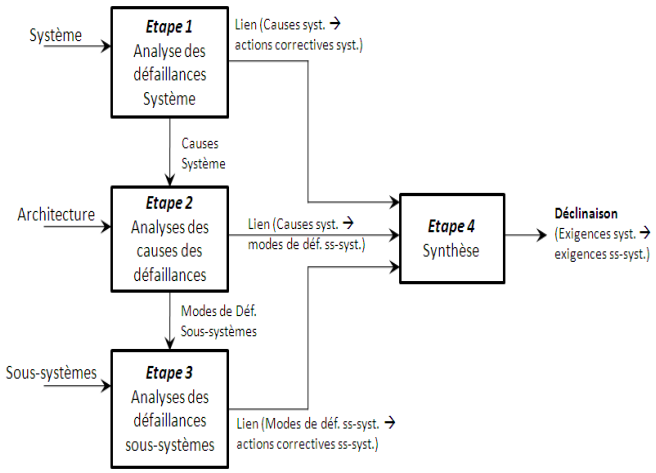

Figure 4. Vue générale de la méthode combinant AMDEC et Arbres de Défaillances

Cette approche respecte le processus de dérivation d'exigences de sûreté expliqué dans [Alexander & al., 2009], où les trois phases fondamentales ont été identifiées :

1. Identification des risques,
2. Analyse des risques,
3. Dérivation des exigences de sûreté de fonctionnement.

4.2 Méthodes utilisées

Pour appliquer cette méthodologie, nous avons choisi d'utiliser des méthodes existantes pour les étapes 1 à 3. Il s'agit de méthodes de sûreté de fonctionnement bien connues et très utilisées que sont : l'Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticités (AMDEC) et l'analyse par Arbre de Défaillances (AdD). L'AMDEC a été sélectionnée car elle est bien adaptée à l'étape 1 et à l'étape 3. L'analyse par AdD est, quant à elle, un bon choix pour réaliser l'étape 2. Il est à noter que d'autres méthodes peuvent être utilisées, du moment qu'elles remplissent l'objectif de l'étape à réaliser.

4.3 Méthodologie de déclinaison

La méthode s'organise en quatre étapes, supposant que le système complexe est composé de plusieurs sous-systèmes. A partir de la définition du système, de ses fonctions, de son architecture, de ses sous-systèmes et de leurs fonctions respectives, la méthode combine des analyses AMDEC et des arbres de défaillances pour définir les exigences de sûreté de fonctionnement système et les informations sur la déclinaison de ces exigences globales en exigences locales associées aux sous-systèmes.

4.3.1 Etape 1 : Analyse des défaillances du système

Dans cette première étape de la méthode, il s'agit de conduire une AMDEC au niveau système. Pour chaque fonction du système, on identifiera les modes de défaillances, leurs causes et leurs effets sur le système (éventuellement en fonction de la phase, l'état ou le mode du système). Puis il faut classifier ces effets et proposer les actions correctives nécessaires appropriées.

Une fois cette première étape achevée, l'AMDEC niveau système aura donc permis d'identifier, d'une part, un ensemble de causes système, ainsi que les criticités liées aux effets de ces causes. D'autre part, en fonction de ces criticités, des actions correctives relatives à ces causes système ont été définies. Ces actions correctives sont alors sources d'exigences de sûreté de fonctionnement à intégrer dans la conception et que le système doit satisfaire pour être sûr. Notamment, elles peuvent se traduire en exigences de sûreté de fonctionnement au travers d'objectifs à respecter en termes de fréquence acceptable (il s'agit dans ce cas d'une action corrective qui porte sur les causes).

On note que des actions correctives peuvent aussi agir sur les effets d'un mode de défaillance, afin de réduire la gravité.

4.3.2 Etape 2 : Analyses des causes des défaillances

A partir des causes système identifiées dans la première étape, la seconde phase consiste à construire les arbres de défaillances qui vont lier les causes systèmes à des modes de défaillances des sous-systèmes. Ainsi, le « top-event » (c'est-à-dire la racine) de chaque arbre est une cause système. Le but est alors de déterminer les causes au niveau sous-système du top-event, en utilisant les opérateurs logiques tel que ET et OU, et sachant que les feuilles des arbres doivent correspondre à des modes de défaillances de sous-systèmes.

A ce stade, la distinction entre un **mode de défaillance** et une **cause** est bien visible. Un mode de défaillance correspond à l'effet par lequel une défaillance est reçue ou observée depuis un point de vue **externe**. Alors qu'une cause représente un événement qui conduit à un (ou plusieurs) mode de défaillance selon un point de vue **interne**.

C'est pourquoi ici, lorsque l'on considère le système et sa composition en sous-systèmes, on essaye de lier les modes de défaillance des sous-systèmes (point de vue externe sur les sous-systèmes), pour arriver, en passant éventuellement par des événements intermédiaires, à des causes système (point de vue interne sur le système). Ces dernières (les causes système) étant liées aux modes de défaillance système à travers l'AMDEC système.

Ainsi, ces arbres (dont un exemple générique est visible figure 5) aide à lier une cause système à un ensemble de modes de défaillance de sous-systèmes.

Ultérieurement, il s'agit d'associer des probabilités aux éléments des arbres d'après les données de fiabilité disponibles et en respectant les objectifs de fiabilité que peuvent imposer les actions correctives de l'AMDEC niveau système. Les données d'entrée pour le calcul des autres probabilités peuvent donc être une combinaison entre des objectifs de fiabilités (niveau système) et des données ou hypothèses concernant les sous-systèmes (c'est-à-dire provenant des feuilles de l'arbre). L'essentiel est de rester cohérent dans la pondération de l'arbre, notamment en respectant les formules basiques de calcul des probabilités vis-à-vis des fonctions logiques. On utilise le terme de « budget » de fiabilité [Lindsay et McDermid, 2002] pour désigner la marge de fiabilité

disponible restant à attribuer aux sous-systèmes dont on n'a aucune donnée à priori.

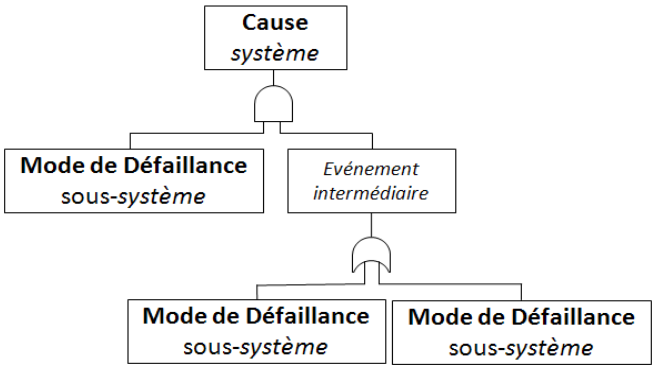

Figure 5. Exemple d'arbre de défaillances

4.3.3 Etape 3 : Analyses des défaillances des sous-systèmes

La troisième étape de la méthode consiste en la réalisation de plusieurs AMDEC, appliquées au niveau sous-système. Les modes de défaillances des sous-systèmes utilisés à l'étape 2 réapparaissent à travers ces AMDEC (par principe des analyses AMDEC). En fait, pour une cause système donnée, il est nécessaire de réaliser une AMDEC pour chaque sous-système dont au moins un mode de défaillance intervient dans l'arbre de défaillances de la cause système en question.
Une fois que cette phase est terminée, l'information utile pour notre méthode provient des relations entre les modes de défaillances des sous-systèmes et les actions correctives du niveau sous-système.

4.3.4 Etape 4 : Synthèse
La quatrième et dernière étape de la méthode est la synthèse qui fournit la déclinaison des exigences de sûreté de fonctionnement niveau système en exigences niveau sous-système.
Dans les étapes précédentes, des liens de 3 types différents ont été créés :

- Entre les causes système et les actions correctives système,
- Entre les causes système et les modes de défaillances sous-système,
- Entre les modes de défaillances sous-système et les actions correctives sous-système.

Ces différents liens permettent de relier les actions correctives système aux actions correctives sous-système. En traduisant les actions correctives par des exigences de sûreté de fonctionnement, nous obtenons des relations entre des exigences de sûreté de fonctionnement niveau système et des exigences de sûreté de fonctionnement niveau sous-système.

5 FORMALISATION UML

Nous allons, dans la section qui suit, proposer une formalisation possible de la méthodologie à l'aide d'UML. Cela revient à définir un méta-modèle liant les différents concepts utilisés, tels que les modes de défaillances, les effets, les causes, les actions correctives, etc.
L'intérêt de cette formalisation est double. Tout d'abord, elle permet de bien poser et montrer la relation entre les différentes notions, ce qui permet de clarifier la compréhension de la méthode. Le deuxième intérêt serait pour une implémentation éventuelle de la méthode dans un outil informatique. Le modèle faciliterait alors l'implémentation de l'outil, notamment concernant la base de données à manipuler.
Nous allons donc présenter le méta-modèle réalisé pour la méthode. Mais pour arriver au modèle complet, il a été nécessaire de formaliser l'AMDEC et les arbres de défaillances.

5.1 Formalisation de l'AMDEC

La formalisation UML de l'AMDEC que nous avons réalisée est visible sur la figure 6. Brièvement, on retrouve le fait qu'une cause implique un (ou plusieurs) mode de défaillance, qui lui-même peut provoquer un effet.

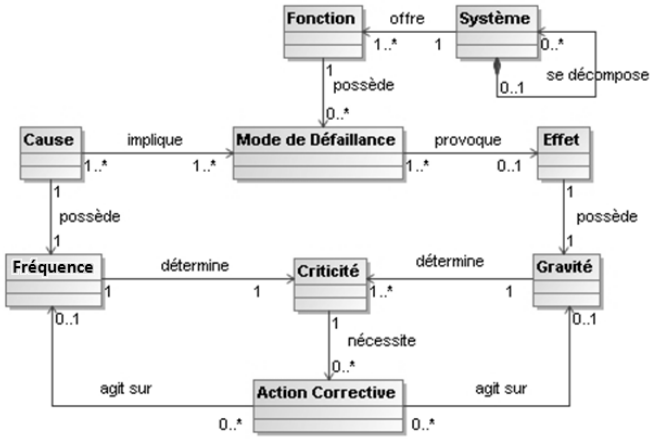

Figure 6 : Formalisation de l'AMDEC en UML

Explication du méta-modèle :
Un système se décompose en sous-systèmes, qui eux-mêmes peuvent être considérés comme des systèmes. Chaque système offre un ensemble de fonctions. Une fonction possède des modes de défaillance. Des causes impliquent ces modes de défaillance. Les modes de défaillance provoquent des effets. Une cause possède une fréquence et un effet possède une gravité. La combinaison de la fréquence d'une cause avec la gravité d'un effet impliqué est source d'une criticité. Selon les cas, une criticité nécessite des actions correctives, qui vont agir sur la fréquence et/ou la gravité.

5.2 Formalisation de l'Arbre de Défaillances

La formalisation de l'arbre de défaillance en UML est donnée par la figure 7. Elle fait intervenir des événements. Ceux-ci peuvent être des causes, mais pas uniquement puisque des événements « intermédiaires » entre les causes d'un système et celles de ces sous-systèmes peuvent être définis. L'autre concept qui apparaît est celui de « porte logique », qui peut être de type ET, OU, voir d'autres types (OU EXCLUSIF, OU PRIORITAIRE, ET PRIORITAIRE,…).

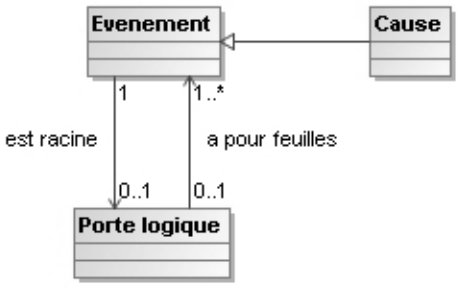

Figure 7 : Formalisation de l'arbre de défaillance en UML

Un événement peut donc être « racine » (c'est-à-dire en sortie) pour une porte logique et/ou feuille (c'est-à-dire en entrée)

pour une porte logique. Les termes « racine » et « feuille » sont utilisés ici pour rappelé que l'on construit un arbre.

5.3 Formalisation de la méthodologie complète

Pour la formalisation de la méthodologie complète, il s'agit d'intégrer les deux diagrammes précédents dans un seul schéma. A cela, il faut ajouter le fait que les actions correctives sont sources d'exigences de sûreté de fonctionnement. Ces exigences sont à prendre en compte pour la conception du système, elles sont donc allouées au système. Au final, le diagramme de la figure 8 formalise tous les concepts de la méthode dans un diagramme de classes UML.

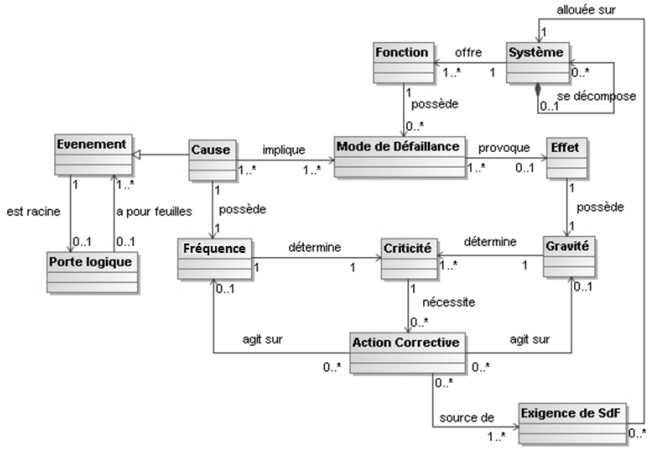

Figure 8 : Formalisation UML des concepts de la méthode AMDEC+AdD

6 CAS D'ETUDE

6.1 Présentation

L'exemple concerne le système de freinage d'un avion. Les phases considérées sont les phases d'atterrissage, de décollage dans le cas d'interruption avant la vitesse V1 (l'avion commence son décollage, l'équipage découvre une anomalie avant la limite de vitesse V1, il décide alors d'interrompre le décollage : freinage d'urgence de l'avion) et la circulation sur les voies de roulage (mode taxi).

Les sous-systèmes qui constituent la fonction de décélération sont les « inverseurs de poussée», les «aérofreins » et les «freins de roue » (voir figure 9).

Les inverseurs de poussée sont associés aux réacteurs et peuvent inverser la direction de la poussée. Ils ne peuvent être utilisés qu'au delà d'une certaine vitesse (autrement les réacteurs réinjectent du gaz chaud et se détériorent).

Les aérofreins sont des surfaces mobiles sur les ailes qui servent à réduire la portance et à augmenter la traînée. En conséquence, ils ralentissent la vitesse de l'avion, en l'empêchant de redécoller et en transférant d'avantage de poids sur les roues. Ils sont efficaces seulement au-dessus d'une certaine vitesse.

Enfin, les freins de roue sont utilisés à toutes les vitesses et sont situés sur les trains d'atterrissage principaux (pas sur le train avant). Ils peuvent être utilisés de façon dissymétrique, pour lutter contre un vent de travers ou pour effectuer des virages serrés.

6.2 Application de la méthode

Nous allons appliquer la méthode sur les aspects «décélération au sol" du système avion, pour identifier les exigences de sûreté et déterminer leur déclinaison au niveau des sous-systèmes.

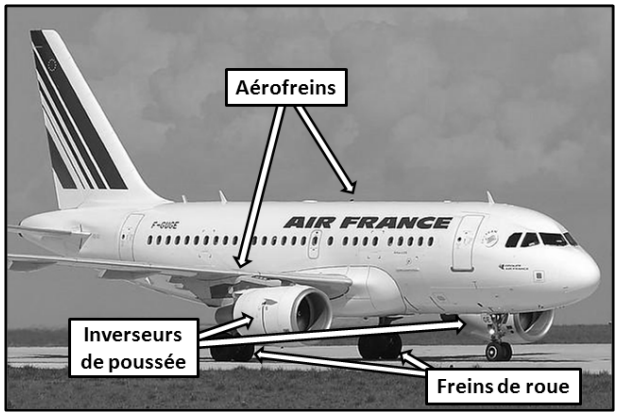

Figure 9. Airbus A320 et les sous-systèmes : inverseurs de poussée», «aérofreins » et freins des roues.

6.2.1 Etape 1: AMDEC système

L'application de l'AMDEC au niveau du système, pour la fonction de décélération est limitée dans le cadre de cet article à un mode de défaillance unique, donné par le tableau 1.

Tableau 1. AMDEC système pour la fonction de décélération au sol

Syst.	Fonction	Mode de défaillance	Phase	Effet	Gravité	Cause	Actions Correctives *(Objectifs de SdF)*
Avion	Freinage de l'avion au sol	Perte de la décélération					
		a) non-annoncée	Atterrissage/ Décollage annulé	L'équipage ne peut pas décelérer l'avion, il en résulte une sortie de piste à haute vitesse	Catastrophique	**Perte non-annoncée de la capacité de décélération**	**Faire en sorte que la fréquence soit<10^{-9}/fh** *(fh. flight hour)*
		b) annoncée	Atterrissage	L'équipage choisi un aéroport approprié, prépare les passagers à une sortie de piste	Hasardeux	**Perte annoncée de la capacité de décélération**	**Faire en sorte que la fréquence soit<10^{-7}/fh**
		c) non-annoncée	Taxi	L'équipage ne peut pas stopper l'avion, il en résulte un contact avec obstacles à faible vitesse	Majeur	**Perte non-annoncée des freins**	**Faire en sorte que la fréquence soit<10^{-5}/fh**
		d) annoncée	Taxi	L'équipage dirige l'appareil évitant les obstacles et demande un remorquage	Sans gravité	**Perte annoncée des freins**	**rien**

Dans ce tableau, nous avons examiné les phases d'utilisation du système parce que les contraintes changent en fonction de ces phases. Cependant, la fréquence n'apparait pas, car elle n'est pas disponible au début de l'étude. Ainsi, les actions correctives sont définies pour garantir une fréquence qui conduit à une criticité acceptable (criticité = fréquence* gravité) pour une gravité donnée.

6.2.2 Etape 2: Construction de l'arbre de défaillance

L'arbre de défaillances de la figure 10 contient la cause racine : « perte non-annoncée de la capacité de décélération ».

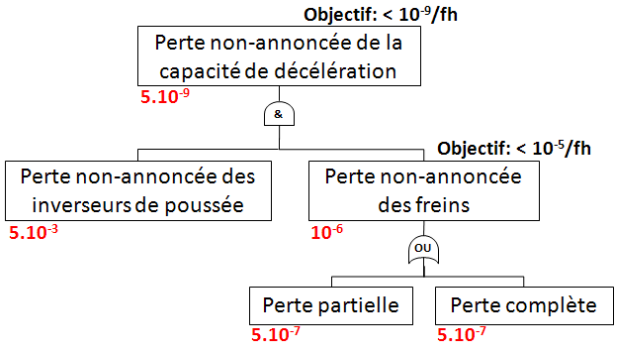

Figure 10. Arbre de défaillance de la perte non-annoncée de la capacité de décélération.

Ainsi, l'arbre est construit progressivement, dans le but d'obtenir les feuilles qui représentent les modes de défaillance des sous-systèmes. Dans cet exemple, nous voyons qu'il est possible qu'un arbre entraîne une « sous-cause » qui correspond à une cause pour un autre mode de défaillance du système, qui est ici: « la perte non-annoncée des freins de roue ».
Pour les calculs de probabilité, nous devons respecter l'objectif de la cause, mais aussi ceux des autres causes survenant dans l'arbre.

6.2.3 Etape : AMDEC Sous-système

Pour cet exemple, nous ne présentons, dans la Tableau 2 que l'AMDEC sous-système du sous-système « freins de roue », avec l'étude de la " sous-fonction « freiner les roues sur le sol sans déraper ». Dans cette étude, nous reconnaissons certains modes de défaillance présents dans l'arbre de défaillance.

6.2.4 Etape 4: Déclinaison et synthèse

La synthèse permet de montrer la décomposition des exigences de sûreté système en exigences de sûreté sous-système. Sans oublier les exigences dues aux interactions (identifiables grâce à l'arbre de défaillance). Nous obtenons que l'exigence de sûreté système « la perte non-annoncée de la capacité de décélération doit avoir une fréquence $<10^{-9}$/fh » est divisée en:
Pour le sous-système « freins de roue » :

- « l'impossibilité d'actionner les freins doit avoir une fréquence $< 5.10^{-7}$/fh »,
- « l'impossibilité d'actionner intégralement les freins doit avoir une fréquence $< 5.10^{-7}$/fh ».

Pour le sous-système « inverseurs de poussée » :

- « l'impossibilité d'actionner les inverseurs de poussée doit avoir une fréquence $< 5.10^{-3}$/fh ».

Pour le niveau « interactionnel » entre les sous-systèmes :

- « La capacité de décélération doit être fournis par les inverseurs de poussée ou les freins ou les deux en même temps ».

7 Conclusion

L'approche proposée dans cet article s'inscrit dans la cadre d'un projet sur la gestion de la sûreté de fonctionnement des systèmes complexes. Elle a pour objectif la définition des exigences de sûreté mais pas uniquement. Elle permet aussi la déclinaison de ces exigences au niveau des sous-systèmes voir composant et d'assurer un lien de traçabilité. L'approche proposée utilise les concepts de processus de l'ingénierie système en se basant sur le standard EIA-632. Cela a permis de proposer une approche multi niveaux et récursive de la prise en compte des exigences de sûreté de fonctionnement du niveau système au niveau composant.
La méthode d'inspire des recommandations des normes ARP-4754 (ED-79/ARP 4754, 1996) et permet de lier les analyses de sûreté à différents niveau du système.
Dans la présentation faite ici la méthode ne s'intéresse qu'aux actions correctives ayant pour objectif la réduction de la fréquence des différents modes de défaillance. La suite du travail en cours prendra en compte les actions correctives permettant de réduire la gravité et la non-détection.

8 References

Alexander R., N. Herbert and T. Kelly (2009), Deriving Safety equirements for Autonomous Systems, 4th SEAS DTC Technical Conference, July 2009.

Avizienis. A, J.-C. Laprie, B. Randell, and C. Landwehr (2004). Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, vol. 1, pp. 11-33, 2004.

Boehm (1999) A Collaborative Spiral Software Process Model Based on Theory W. August 11, 1994.

Buzzatto J.L (1999). Failure mode, effects and criticality analysis (FMECA) use in the Federal Aviation Administration (FAA) reusable launch vehicle (RLV) licensing process. Digital Avionics Systems Conference, 1999.Proceedings 18th. vol.2 10/24-29/1999. Location: St Louis, MO, USA.

Chavalarias D., Bourgine P., Perrier E., Amblard F., Arlabosse F., Auger P., Baillon J.-B., Barreteau O., Baudot P., Bouchaud E. (2008). French Roadmap for complex Systems 2008-2009, French National Network for Complex Systems (RNSC), Paris Ile-de-France Complex Systems Institute (ISC-PIF) and IXXI, "Entretiens de Cargèse 2008", 2008.

Forsberg, Kevin and Harold Mooz (1995) "Application of the "Vee" to Incremental and Evolutionary Development," Proceedings of the Fifth Annual International Symposium of the National Council on Systems Engineering, St. Louis, MO, July 1995.

Goguen. J and C. Linde (1993). Techniques for requirements elicitation. In 1st IEEE International Symposium on Requirements Engineering, pages 152-164, San Diego, 4-6th January 1993.

Gotel. O. C. Z. and C. W. Finkelstein (1994). "An analysis of the requirements traceability problem," in International Conference on Requirements Engineering, 1994, pp. 94–101.

Guillerm. R, Demmou. H and Sadou. N (2009). System engineering approach for safety management of complex systems. Proceedings of European Modeling and simulation (ESM'2009). October 26-28, 2009 Leicester, United Kingdom.

Tableau 2. AMDEC du sous-système « freins de roue »

Syst.	Fonction	Mode de défaillance	Phase	Cause	Effet	Actions Correctives *(Objectifs de SdF)*
Freins des roues	Freiner les roues sur le sol sans dérapage	Perte complète	Au sol	Impossible d'actionner les freins	Les roues ne sont pas freinées	Faire en sorte que la fréquence soit < 5.10^{-7}/fh
		Perte partielle	Au sol	Impossible d'actionner intégralement les freins	Les roues ne sont pas intégralement freinées	Faire en sorte que la fréquence soit < 5.10^{-7}/fh

Guillerm. R, Sadou. N, Demmou. H (2010) Information model for model driven design of complex system based on system engineering approach International Conference onComplex Systems Design and Management (CSDM 2010), Paris (France), 27-29 Octobre 2010, pp.99-111

Juristo. N, A. M. Moreno, and A. Silva (2002) "Is the European Industry Moving Toward Solving Requirements Engineering Problems?" IEEE Software, vol. 19, no. 6, pp. 70–77, 2002.

Komi-Sirvio. S and M. Tihinen, "Great Challenges and Opportunities of Distributed Software Development – An Industrial Survey." in Proceedings of the Fifteenth International Conference on Software Engineering & Knowledge Engineering (SEKE'2003), 2003, pp. 489–496.

Lee. W.S, D. L. Grosh, F. A. Tillman, C. H Lie (1985). "Fault tree analysis, methods, and applications - A review", IEEE Transactions on Reliability, August 1, 1985; ISSN 0018-9529; r-34, page 194- 203.

Lindsay P. A. et J. A. McDermid (2002). Derivation of safety requirements for an embedded control system, Systems Engineering, Test and Evaluation Conference, Sydney, 29-30 Octobre 2002.

Magee C. and de Weck O. L (2004). Complex System Classification, Fourteenth Annual International Symposium of the International Council on Systems Engineering (INCOSE), Toulouse, France, June 20-24, 2004.

Meinader, J.-P. (2002) Le métier d'intégration de systèmes, Edition Hermès-Lavoisier (540p), 2002.

Sahraoui. A.-E.-K, D. Buede, A. Sage (2004). "Issues in systems engineering research," INCOSE congress, Toulouse, 2004.

Sahraoui. A.-E.-K. (2005). "Requirements Traceability Issues: Generic Model, Methodology and Formal Basis." International Journal of Information Technology and Decision Making, vol. 4, no. 1, pp. 59–80, 2005.

Sommerville. I (2006). Software Engineering: (Update) (8th Edition) (International Computer Science). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.

Spitzer Cary R., Avionics Development and Implementation, 2nd Edition, CRC Press, 2007.