SYNTHESIS : une méthodologie outillée d'évaluation et d'optimisation des performances de sûreté de fonctionnement d'un système complexe

André Leblond, Michel Batteux, Antoine Rauzy

Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

SYNTHESIS : une méthodologie outillée d'évaluation et d'optimisation des performances de sûreté de fonctionnement d'un système complexe

SYNTHESIS : a tooled method for evaluating and optimizing safety performances of a complex system

André Leblond
Altarica Association
Les Essarts le Roi, France
andreleblond@orange.fr

Michel Batteux
IRT SystemX
Palaiseau, France
michel.batteux@irt-systemx.fr

Antoine Rauzy
NTNU-MTP
Trondheim, Norway
antoine.rauzy@ntnu.no

***Résumé* — SYNTHESIS permet d'évaluer, puis d'optimiser les performances de sûreté d'un système complexe. Elle passe par la décomposition en briques fonctionnelles des composants. Un algorithme dédié est utilisé pour la génération d'indicateurs qualitatifs et quantitatifs représentatifs de ces performances.**

Mots-clés — coupe minimale, composant, CSA, évaluation de la sûreté de fonctionnement, MBSA, synthèse de coupes

***Abstract* — SYNTHESIS makes it possible to assess and then optimize safety performances of a complex system. It involves the decomposition into functional bricks of the components. A dedicated algorithm is used to generate qualitative and quantitative indicators representative of these performances.**

Keywords—minimum cutset, component, compositional safety assessment (CSA), evaluation of safety performances, MBSA, cutsets synthesis

I. INTRODUCTION ET CONTEXTE

La démonstration de la sûreté d'un système complexe est une tâche qu'il est préférable de commencer lors des étapes initiales de définition dudit système (« early validation »). L'impact d'une correction d'erreur sur les coûts et les délais est en effet d'autant plus grand que cette erreur est détectée plus tard. Ceci est particulièrement vrai pour les systèmes avioniques hautement intégrés.

Par ailleurs, les pratiques courantes du métier sont fondées sur des tâches manuelles, fortement sujettes à erreur, qui peuvent au final augmenter les coûts de certification. Ceci explique l'apparition, à partir des années 1990, des techniques d'« évaluation de la sûreté à base de modèles » (Model-Based Safety Assessment, MBSA) dans les milieux universitaires et industriels.

L'utilisation de modèles est particulièrement recommandée lors de la phase d'« early validation » mentionnée ci-dessus. En permettant à l'équipe de développement de découvrir rapidement certains problèmes, elle augmente son degré de confiance dans la solution choisie. Ces modèles doivent pouvoir être conçus et mis à jour facilement, de manière à écarter au moindre coût les solutions fautives. Le développement du système peut ensuite se poursuivre d'une façon moins chaotique que si l'on avait procédé autrement.

L'un des critères utilisés pour classifier les techniques de MBSA concerne la relation existant entre les modèles utilisés pour l'analyse de sûreté et les modèles issus du processus principal de développement. On peut distinguer deux approches : les premiers sont des *extensions* des modèles utilisés pour l'ingénierie, les seconds sont des *modèles dédiés* venant s'ajouter aux éventuels modèles système.

Un exemple de la première approche MBSA est la construction d'un « Modèle Système Étendu avec Injection de Pannes » (Extended System Model with Failure Injection) [9, 10, 11]. Les experts sûreté de fonctionnement reçoivent de la part de l'équipe de développement, des modèles d'ingénierie système écrits dans des langages du type SCADE ou Matlab Simulink. Ces modèles sont ensuite enrichis par des « modèles de défaillances » élémentaires qui viennent s'insérer dans les flux d'information du modèle principal. Les propriétés requises par l'analyse de sûreté sont alors vérifiées par simulation en injectant aléatoirement des fautes.

L'avantage principal de l'approche « modèle étendu » est la cohérence reliant par construction les analyses de sûreté et le processus principal de développement. Cependant, cette cohérence dépend de l'existence de modèles système formels et exécutables. Elle n'est donc applicable qu'aux étapes avancées du processus d'ingénierie (e.g. vers la fin de la PSSA ou de la SSA), quand la conception du système est suffisamment détaillée, stable et mature. À ce stade, le coût

d'un changement est élevé, et l'analyse de sûreté « perd l'occasion » d'influencer et de participer à la conception d'une manière optimale. De plus, les approches par injection de fautes ne couvrent par définition que partiellement les problèmes éventuels et sont extrêmement coûteuses en temps de calcul [2]. Enfin, les niveaux de granularité pertinents pour la conception du système et pour l'analyse de sûreté ne sont pas forcément les mêmes. De nombreux détails, sans intérêt pour l'analyse de sûreté, sont en général inclus dans le modèle système.

La deuxième approche MBSA, fondée sur l'utilisation de modèles dédiés à la sûreté, distincts des modèles système, allège sensiblement les difficultés mentionnées ci-dessus. Elle comprend notamment les techniques de « Compositional Safety Assessment (CSA) », où l'évaluation de la sûreté de fonctionnement pour un système complexe est décomposée en tâches élémentaires d'évaluation de la sûreté des composants individuels qui constituent le système. L'expert métier choisit les composants qu'il juge significatifs pour son analyse, et décrit leur comportement et leurs interfaces au niveau de détail juste nécessaire. Des techniques et notations MBSA avancées, telles que HiP-HOPS [3, 4], FPTN [5], et AltaRica [6, 7], s'inscrivent dans ce contexte. Le comportement des composants modélisés est vu comme une relation de dépendance entre d'une part l'état des sorties, et d'autre part l'état des entrées et le statut actif/inactif des défaillances internes (« Failure Logic Modeling », FLM).

Une des faiblesses de cette approche est que la cohérence entre modèles pour l'ingénierie et modèles pour l'analyse de sûreté n'est pas garantie, elle doit être maintenue tout au long du processus de développement. Il existe aussi un fort besoin d'établir et de conserver des relations de « raffinement » et d'« abstraction » entre modèles de sûreté de niveaux de détail différents. Des modèles dysfonctionnels de haut niveau sont requis pour l'« early validation ». Par la suite les composants constituant ces modèles devront être « raffinés », en laissant ouverte la possibilité de réinjecter les dysfonctionnements détaillés dans les modèles composants primitifs (« abstraction ») et de définir ainsi des défaillances « composants » [2].

II. SOMMAIRE DE LA DEMARCHE ET OBJECTIFS

L'objectif général de cet article est de décrire la méthode outillée SYNTHESIS d'évaluation et d'optimisation des performances de sûreté de fonctionnement, applicable aux stades amont (« early validation ») de la conception d'un système complexe. La démarche proposée est une approche « modèles dédiés », qui consiste à effectuer, autant de fois qu'il est nécessaire, un cycle comprenant deux étapes : la construction (ou mise à jour) d'un modèle système, suivie de l'évaluation des performances de sûreté. À l'issue du premier cycle, si les objectifs ne sont pas tenus, l'expert définit pour son système une (ou des) barrière(s) de sûreté supplémentaire(s). Le cycle à deux étapes mentionné ci-dessus est alors réitéré.

La première étape consiste, pour chacune des architectures envisagées pour le système et chaque évènement redouté, à construire un modèle dysfonctionnel « dédié à la sûreté » à l'aide d'un outil adapté, par exemple Cécilia (Dassault Aviation) ou AltaRica 3.0 [7]. L'expert modélise le comportement dysfonctionnel des composants, puis combine ces modèles pour aboutir au modèle système.

La construction d'un tel modèle soulève des difficultés, notamment la définition, lors des phases préliminaires de l'étude, des *modèles des composants* et des comportements dysfonctionnels associés. À ce stade, la structure physique des composants n'est pas encore figée. En l'absence d'AMDEC détaillées, les modes de défaillance physiques ne sont pas non plus clairement identifiés. Les données dysfonctionnelles dont dispose l'expert pour ces composants se réduisent souvent à des estimations de leurs taux de défaillance. Les logiques de défaillances (FLM) à établir pour chacun d'entre eux doivent donc s'appuyer sur d'autres sortes de défaillances.

La méthode de construction retenue pour SYNTHESIS passe par la décomposition des composants en *opérateurs fonctionnels interconnectés* comprenant des *défaillances génériques*, abstraction faite des structures physiques des composants. Il s'agit donc d'un « raffinement » du niveau « composants » en un niveau « opérateurs fonctionnels » plus fin. Les logiques de défaillances associées aux opérateurs sont ensuite établies (FLM). Les modèles des composants résultent de l'assemblage de ces logiques partielles. L'outil de modélisation permet alors de générer le jeu de coupes (ou de séquences) minimales associées, que nous nommerons par la suite *coupes minimales individuelles* ou *fonctionnelles.* Elles sont des combinaisons de *défaillances d'opérateurs* produisant l'évènement redouté. Nous les nommerons aussi *indicateurs qualitatifs détaillés*.

La deuxième étape de la méthode consiste, à partir des résultats fournis par la première, à évaluer les performances de sûreté du système modélisé. Dans une approche classique, des probabilités sont associées aux défaillances intervenant dans les coupes, et l'outil permet de calculer la probabilité de l'évènement redouté par heure de mission.

Malheureusement, dans le cas qui nous occupe, les coupes minimales sont constituées de données abstraites, non quantifiées (défaillances d'opérateurs). Leur volume peut aussi être important. Des jeux de coupes comprenant plusieurs milliers, voire dizaines ou centaines de milliers d'éléments sont fréquemment rencontrés. Nous ne savons notamment pas identifier les composants les plus critiques, pouvant provoquer un évènement redouté à eux seuls ou en association avec un autre composant.

La *synthèse des coupes fonctionnelles* en *coupes globales* que nous proposons pour surmonter cette difficulté est la principale innovation décrite dans cet article. Elle est complémentaire de la construction du modèle et justifie son existence. Elle consiste en un retour (« abstraction ») du niveau détaillé « opérateurs fonctionnels » vers le niveau « composants », qu'elle enrichit de nouvelles données tout en assurant la traçabilité entre niveaux.

Ce processus permet ainsi d'exprimer un grand nombre de coupes minimales détaillées sous une forme synthétique, quantifiée, et facilement interprétable (*indicateurs synthétiques*, i.e. *coupes globales*). Ces indicateurs sont étroitement liés au niveau de sûreté couramment atteint par le modèle. L'ajout d'une barrière de sûreté aura par exemple un impact significatif sur ces indicateurs.

Dans ce cadre, le deuxième objectif de l'article est d'une part de décrire la démarche de construction appliquée à notre exemple (première étape) et d'autre part de présenter les résultats de la synthèse de coupes en termes d'indicateurs (deuxième étape). Le troisième objectif de l'article est de montrer sur notre exemple comment les fonctionnalités du

système peuvent évoluer et être optimisées du point de vue de la sûreté de fonctionnement à l'aide de ces indicateurs.

III. LE SYSTEME EXEMPLE

Le système exemple considéré dans l'article est constitué par quatre « modules » COM-MON interconnectés, embarqués sur un avion (Fig. 1, deux composants par module). Le premier étage de modules, constitué par *FMCOM1-FMMON1* et *FMCOM2-FMMON2*, a pour entrées les données générées par trois composants *ADIRS*, chacun d'eux regroupant trois fonctions : Air Data Computer, centrale inertielle et GPS. Ce premier étage génère en sortie les « angles commandés » *Angles1* et *Angles2* de l'avion par rapport aux trois axes. À tout instant, les trois angles commandés sont ceux qu'il faut faire prendre à l'avion pour suivre une trajectoire précalculée.

Les sorties du premier étage sont présentées en entrée des modules *FGCOM1-FGMON1* et *FGCOM2-FGMON2* du deuxième étage. Les sorties sont les « Flight Directors » *FlightDirectors1* et *FlightDirectors2* à afficher sur des écrans. Les « Flight Directors » sont des représentations graphiques de l'écart existant entre les trois angles réels de l'avion et les trois angles commandés. Le pilote dirige son avion en tendant à réduire à zéro ces écarts angulaires.

L'ensemble des quatre modules doit globalement tenir des exigences sévères de sûreté, pour l'intégrité des Flight Directors générés, et pour la disponibilité globale de ces données. Le principe retenu pour tenir ces objectifs est la surveillance mutuelle exercée par chaque couple de composants *COM* et *MON*. Des alarmes visibles pour le pilote sont générées par ces composants en cas d'incohérence.

Dans le contexte de cet article, l'avion est conduit par le pilote en utilisant la voie 1. En cas de perte de la voie 1 (i.e. « Flight Directors voie 1 perdus OU alarme générée par l'un des deux COM-MON constituant la voie 1 »), le pilote utilise la voie 2. L'évènement redouté d'intégrité est « Flight Directors corrompus sur la voie 1 ET pas d'alarme sur cette voie 1 ». L'évènement redouté de disponibilité est « (Flight Directors perdus sur la voie 1 OU Alarme sur la voie 1) ET (Flight Directors perdus sur la voie 2 OU Alarme sur la voie 2) ». Ces deux événements redoutés sont de sévérité « Hazardous » (Proba < 1E-7). Dans la suite de cet article nous ne considérerons que l'événement redouté d'intégrité.

IV. CONSTRUCTION DU MODELE ET COUPES MINIMALES

Dans ce paragraphe nous présentons la démarche de construction tout en nous appuyant sur le système exemple. Le modèle système est constitué par un ensemble de *composants modèles* interconnectés, chacun d'eux étant l'image d'une sous-partie du système physique, que nous nommerons *composant physique*. Dans la suite de cet article, et sauf exception, les composants modèles et physiques seront désignés par le terme générique « composant » qui selon le contexte représentera l'un ou l'autre aspect.

La construction du modèle passe par la décomposition des composants en opérateurs fonctionnels interconnectés, et par l'établissement des logiques de défaillances de ces opérateurs, abstraction faite des structures physiques des composants. Il s'agit donc d'un *raffinement* du niveau « composants » en un niveau « opérateurs » détaillé.

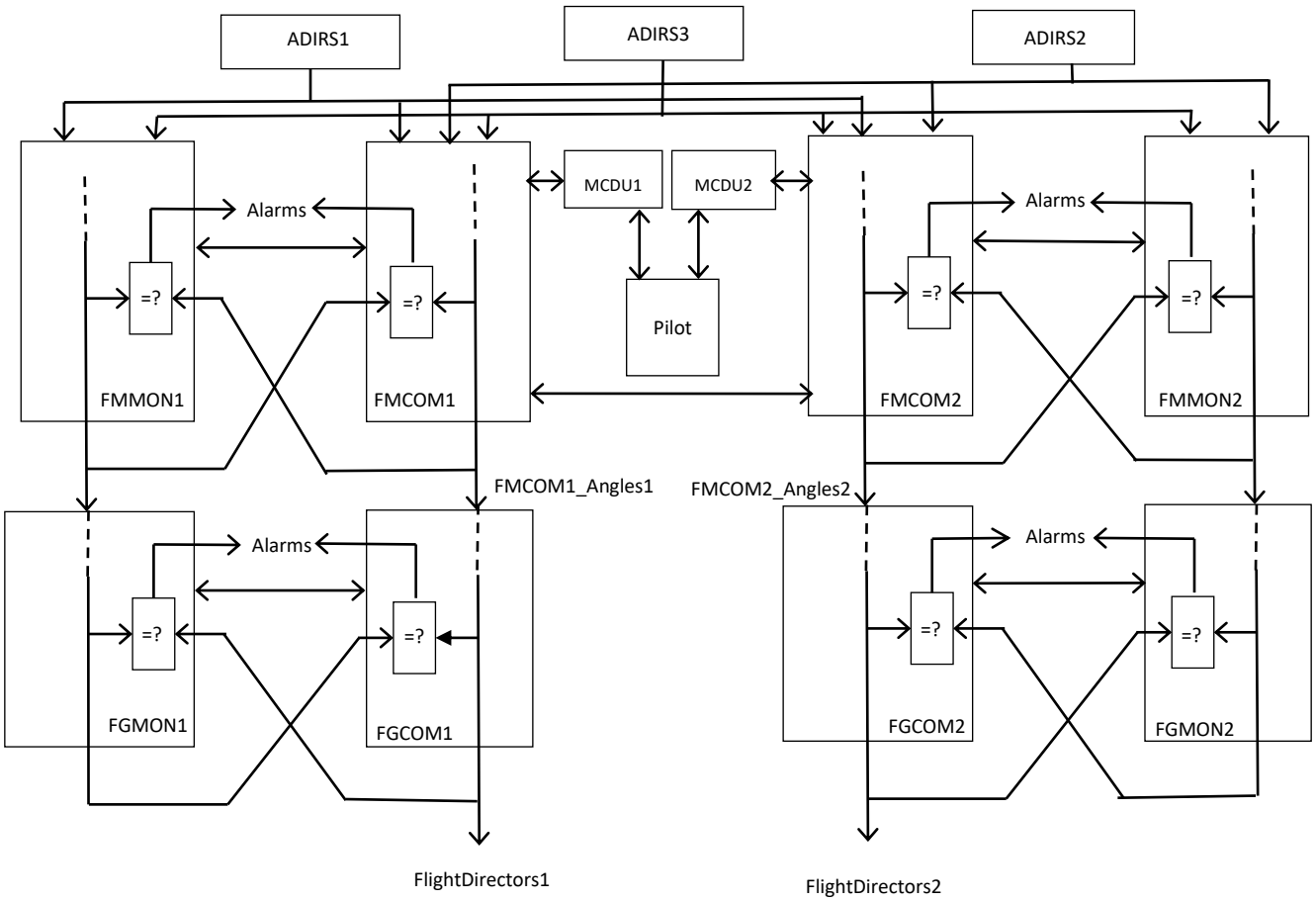

Fig.1. Quatre COM-MON interconnectés

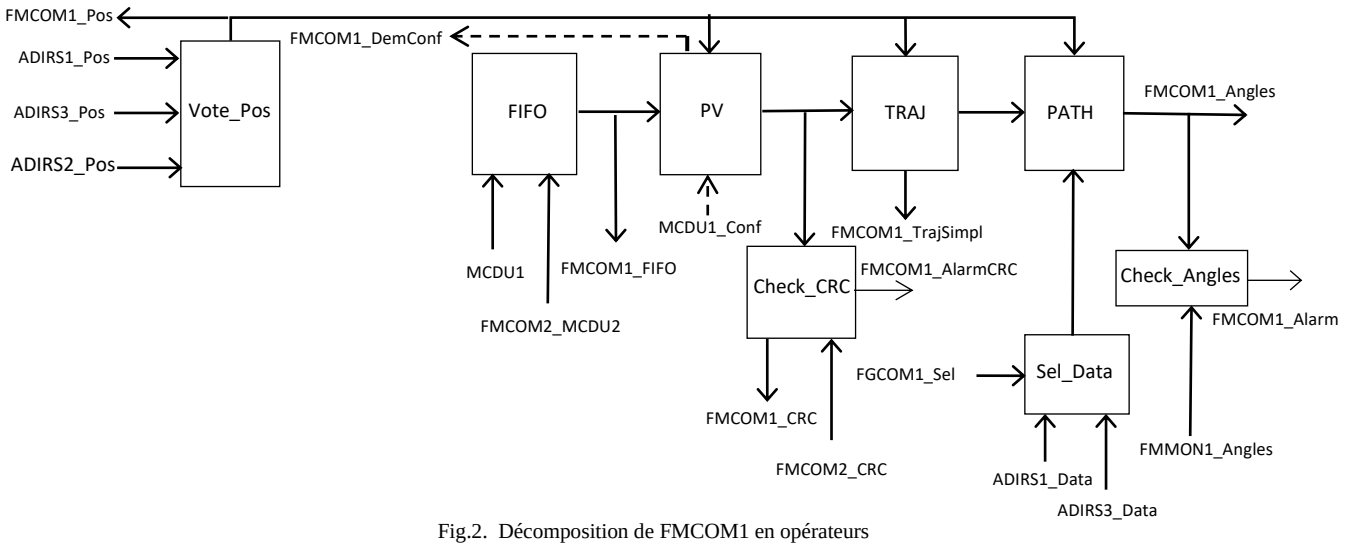

Fig.2. Décomposition de FMCOM1 en opérateurs

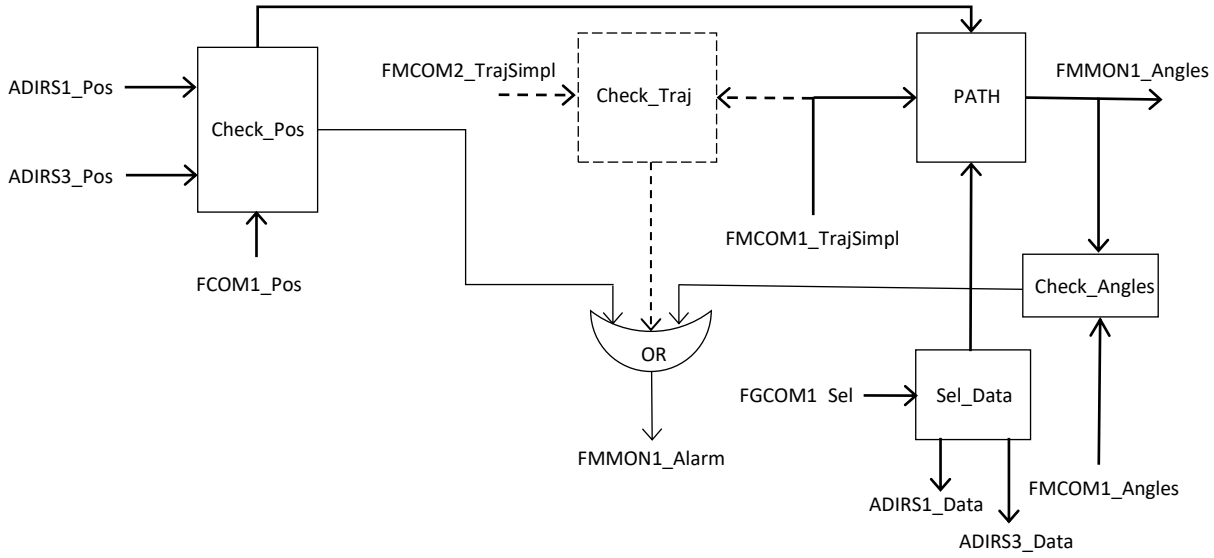

Fig. 3. Décomposition de FMMON1 en opérateurs

A. Opérateurs fonctionnels sous-jacents

À défaut de structure physique pour un composant, on dispose, de manière non formelle, d'informations de nature diverse : orales, textuelles, graphiques ou autres. Pour *FMCOM1*, on sait par exemple que le plan de vol est calculé à partir de la position de l'avion et d'informations saisies par le pilote sur *MCDU1* ou *MCDU2*. La trajectoire (segments plus arcs de cercle) est construite à partir du plan de vol et de la position. Les angles commandés sont calculés à partir de la trajectoire. La position doit résulter de l'ensemble des trois *ADIRS* redondants, et les angles commandés issus de *FMCOM1* doivent être comparés aux angles analogues issus de *FMMON1*.

La Fig. 2 est le résultat de cette réflexion. C'est une *structure fonctionnelle et non physique*. Les pointillés correspondent à des flux de données internes de *FMCOM1*, non présents dans sa version de base. On notera l'opérateur *Check_CRC*, qui calcule le CRC du plan de vol de *FMCOM1*, et le compare avec cette même donnée issue de *FMCOM2*. En cas d'incohérence une alarme est générée.

De même, une autre alarme est éventuellement générée suite à la comparaison des angles commandés générés en interne de *FMCOM1* et de ceux issus de *FMMON1*. Ces deux alarmes sont résumées en une seule sur la Fig. 1.

À titre d'exemple supplémentaire, la Fig. 3 montre la découpe fonctionnelle de *FMMON1*. Les positions issues de *ADIRS1* (*ADIRS2* pour la voie 2) et de *ADIRS3* sont comparées, produisant une alarme éventuelle. Une autre alarme éventuelle est générée par comparaison entre les angles commandés issus de *FMCOM1* et ceux générés en interne de *FMMON1*. L'entrée *FMCOM1_TrajSimpl* est un « squelette » de la trajectoire calculée par *FMCOM1* (trajectoire simplifiée).

Cette méthode de décomposition fonctionnelle des composants constitue un point-clé de la démarche SYNTHESIS. Les schémas ainsi produits doivent être approuvés et pris en compte par le concepteur du système. Ils constituent des spécifications partielles de ces composants.

B. *Défaillances pour les opérateurs et composants modèles*

Dans le contexte de notre article, nous considérons deux sortes d'information : *continue* ou *discrète*. Les informations continues peuvent prendre une plage de valeurs, elles sont codées sur trois états : *OK* (valeur *correcte*), *KO* (valeur *corrompue*) et *NO* (information *absente*). Les informations discrètes ne prennent qu'un nombre fini (et généralement réduit) de valeurs. Dans l'article, et sans perte de généralité, les informations discrètes sont uniquement booléennes (*true* et *false*).

À une sortie continue d'opérateur sont associées les défaillances génériques *perte partielle* et *corruption*. La défaillance *perte partielle* force la sortie à *NO*. La défaillance *corruption* force la sortie, si elle est présente, à *KO*. À une sortie booléenne d'opérateur sont associées les défaillances *collage à false* et *collage à true*. Chaque opérateur fonctionnel comprend donc des défaillances *perte partielle*, *corruption*, *collage à false*, *collage à true*, correspondant à ses sorties continues ou booléennes. Notons que cette liste n'est pas exhaustive et que d'autres choix sont possibles (e.g. la génération intempestive d'une sortie), qui n'altèrent en rien la démarche. Les *défaillances fonctionnelles* associées à un *composant modèle* sont toutes celles associées aux opérateurs qui lui sont sous-jacents.

Il est recommandé de définir, en plus de cela, une défaillance *perte totale* attachée globalement à un composant modèle avec ses opérateurs sous-jacents. Elle provoque à elle seule la perte simultanée de toutes les sorties continues de ces opérateurs, et le collage à une valeur par défaut de toutes les sorties booléennes. En AltaRica, cette défaillance peut être modélisée par une « synchronisation ». Il n'y a pas d'inconvénient à ce qu'un composant comporte plusieurs défaillances *pertes totales*.

C. *L'établissement des logiques de défaillances*

À chaque opérateur est associée une *logique de défaillances* : les états des sorties sont des fonctions booléennes des états des entrées et du status actif/inactif des défaillances internes (FLM). L'opérateur *Vote_Pos* de *FMCOM1* est un voteur à trois entrées *ADIRSi_Pos*. Si les trois entrées sont présentes, et que l'état *OK* ou *KO* de *ADIRS1_Pos* est égal à celui de *ADIRS2_Pos* ou à celui de *ADIRS3_Pos*, *ADIRS1_Pos* est sélectionnée, autrement *ADIRS3_Pos* est sélectionnée. Si deux entrées sont présentes, *ADIRS1_Pos* est sélectionnée si elle est présente. Autrement *ADIRS3_Pos* est sélectionnée. De même, si une seule entrée est présente, *ADIRS1_Pos* est sélectionnée si elle est présente, autrement *ADIRS3_Pos* est sélectionnée. L'entrée sélectionnée peut elle-même être corrompue ou perdue en sortie si l'une des défaillances *corruption* ou *perte partielle* de *FMCOM1_Pos* est active.

Cette logique résulte de discussions informelles autour de ce voteur. D'autres choix sont possibles, notamment dans les cas où une ou plusieurs entrées sont absentes. Les coupes minimales fonctionnelles générées par la suite dépendent des hypothèses retenues. Ces logiques doivent elles aussi être approuvées et prises en compte par le concepteur du système. Elles constituent, au même titre que les réseaux d'opérateurs sous-jacents, des spécifications partielles des composants.

D. *Coupes fonctionnelles et problématique de complexité*

Pour mémoire, l'événement redouté considéré est « Flight Directors corrompus sur la voie 1 ET pas d'alarme sur cette voie 1 ». L'outil de modélisation permet de générer les coupes minimales fonctionnelles associées, elles sont résumées dans la partie gauche de la Table 1 (modèle de base : pas de Check_Pil, pas de Check_Traj). Elles constituent l'*indicateur qualitatif détaillé* du niveau de sûreté du système modélisé.

TABLE 1. TABLEAU RECAPITULATIF DES COUPES

	No Check_Pil No Check_Traj			**Check_Pil No Check_Traj**			**Check_Pil Check_Traj**		
Order	**Min Fonct Cut**	**Min Glob Cut**	**IQ**	**Min Fonct Cut**	**Min Glob Cut**	**IQ**	**Min Fonct Cut**	**Min Glob Cut**	**IQ**
1	4	4	2,0E-5		1	5,0E-6			
2	44	10	1,1E-09	44	12	1,1E-09	41	19	1,5E-09
3	45			61			71		
4	3			15			13		
5							47		
6							48		
TOTAL	**96**	**14**	**2,0E-5**	**120**	**13**	**5,0E-6**	**220**	**19**	**1,5E-09**

Pour notre exemple, le nombre de ces coupes est relativement réduit, mais pour certains modèles il peut atteindre la dizaine voire la centaine de milliers. Elles sont souvent d'ordre élevé (6 ou 7, voire davantage). N'étant pas quantifiées, on ignore leur poids relatif, ce qui interdit de les éliminer sur ce critère. Pour toutes ces raisons, ces coupes fonctionnelles ne sont pas exploitables directement, mais doivent au préalable être traitées et réduites.

C'est tout l'objet de la *synthèse des coupes fonctionnelles* en *coupes globales*, la principale innovation décrite dans cet article. Elle est complémentaire de la démarche de construction du modèle. Avant de la décrire, nous introduirons la notion de *type* pour les défaillances.

V. DEFAILLANCES ET COUPES PARTIELLES TYPEES

A. *Les défaillances composants physiques typées*

D'une manière générale, on peut considérer pour un composant physique une défaillance globale qui regroupe l'ensemble des pannes matérielles, à laquelle on associe le taux Λ de défaillance total du composant. Dans cet article on adopte une classification plus fine, avec deux groupes distincts de défaillances : *pertes totales* et *autres dysfonctionnements*. Les *types* respectifs de ces deux groupes seront notés *L* et *F*.

Les défaillances *pertes totales* (type *L*) d'un composant physique provoquent sa « passivation » : les sorties sont absentes ou figées à des valeurs par défaut. Entrent notamment dans *pertes totales* les pannes détectées par les tests en ligne BITE, dont la sanction est l'arrêt du composant, et les pannes d'alimentation provoquant son extinction. L'AMDEC du composant, si elle existe, permet d'affecter une valeur Λ_L à la probabilité d'occurrence par heure de *pertes totales*, le composant étant initialement exempt de pannes. Dans le cas contraire, un ordre de grandeur $\Lambda_L = 90\%\Lambda$ peut être utilisé.

Les défaillances *autres dysfonctionnements* (type *F*) regroupent tout ce qui n'est pas *pertes totales*. Elles comprennent donc les *générations d'erronés*, les *pertes de fonctionnalités*, etc. En termes d'AMDEC, il s'agit des pannes physiques mentionnées dans la liste dont les effets ne sont pas « perte du composant ».

B. Les défaillances fonctionnelles typées

Par analogie avec les défaillances typées des composants physiques nous définissons deux groupes typés de défaillances pour les *composants modèles*.

Les défaillances *pertes totales* d'un composant modèle (type *L*) comprennent la (ou les) défaillance(s) dont chacune provoque à elle seule la perte de toutes les sorties continues des opérateurs sous-jacents, et le collage à une valeur par défaut de toutes les sorties booléennes.

Les défaillances *autres dysfonctionnements* d'un composant modèle (type *F*) comprennent les défaillances *corruptions*, *collages à false*, *collages à true*, *pertes partielles*, *etc.* associés aux sorties des opérateurs sous-jacents du composant.

C. Les coupes partielles typées

Une *coupe partielle* d'une coupe minimale fonctionnelle sur un composant modèle *C* est constituée de l'ensemble des défaillances de la coupe situées sur *C*. L'état des sorties d'un composant modèle *C* quand une défaillance de type *L* est active reste le même (absentes ou collées à des valeurs par défaut) quand des défaillances supplémentaires dans *C* sont activées. Il n'y a donc en fait que deux sortes de coupes partielles :

- Un ensemble de défaillances fonctionnelles de type *F* : coupe partielle de type *F*.
- Un ensemble réduit à une seule défaillance fonctionnelle de type *L* : coupe partielle de type *L*.

Il ne s'agit pas là d'une condition nécessaire pour pouvoir appliquer la synthèse. L'algorithme décrit ci-dessous est en fait plus général, et capable de traiter des coupes partielles qui sont des mélanges de types. Mais ceci dépasse le cadre de cet article.

VI. ÉVALUATION DES PERFORMANCES DE SURETE

Ce paragraphe contient la description de la deuxième étape de la méthode SYNTHESIS, i.e. la synthèse des coupes fonctionnelles en coupes globales. Son application au système exemple sera décrite au § VII. Cette synthèse est complémentaire de la construction du modèle, et justifie son existence. Elle consiste à générer, à partir des coupes minimales individuelles produites par la première, des *indicateurs qualitatifs et quantitatifs synthétiques* (coupes globales) des performances de sûreté qui caractérisent ces modèles système et les évènements redoutés associés. Les comparaisons entre architectures du point de vue de la sûreté sont alors possibles et s'appuieront sur ces indicateurs.

A. Description de la synthèse en coupes globales

1) Motivation

En pratique, on observe de nombreux cas de coupes minimales couvrant les mêmes composants, les types de coupes partielles étant identiques sur ces composants. Ces coupes minimales sont dites *similaires*. On observe également (cf. § VII.C et § VII.D) que la suppression d'une coupe minimale suite à une évolution du modèle s'accompagne fréquemment de la suppression d'autres coupes minimales similaires.

D'où l'idée de grouper les coupes minimales en *coupes globales*, ou *coupes synthétiques*, ou encore *coupes composants*. Une coupe globale est une combinaison de défaillances composants typées. Elle groupe un ensemble de *coupes minimales sous-jacentes similaires*, i.e. couvrant les mêmes composants avec cohérence des types. Dans de nombreux cas, une évolution du modèle provoquera la suppression de une ou plusieurs coupes globales, et de toutes les coupes sous-jacentes associées (cf. § VII.C et § VII.D).

2) Principe de génération des coupes globales

Chaque coupe minimale est convertie en une coupe globale. Une *coupe partielle* de type *F* ou *L* d'un composant *C* est convertie en une *défaillance physique* F_C ou L_C. La coupe globale est constituée par l'ensemble des défaillances composants typées correspondant à la coupe minimale initiale.

Les coupes globales redondantes sont supprimées au fur et à mesure de leur génération. En option, les coupes globales non minimales peuvent également être supprimées. L'ensemble des coupes globales, ou globales minimales, constitue l'*indicateur qualitatif synthétique* du niveau de sûreté présentement atteint par le modèle système.

3) Quantification des coupes globales

On suppose l'absence de défaillances latentes de composants physiques en début de mission. Les taux de défaillance relatifs aux défaillances composants typées sont fournis par l'utilisateur. Dans ce cadre, la probabilité associée à une telle défaillance est le produit du taux de défaillance et de la durée en heures de la mission.

L'*indicateur quantitatif synthétique associé à une coupe globale* (minimale ou non minimale) est le produit des probabilités relatives aux défaillances composants de cette coupe globale. L'*indicateur quantitatif synthétique associé à l'ER* est la somme des indicateurs associés aux coupes globales.

Notons bien qu'il ne s'agit ni de la probabilité d'occurrence de l'événement redouté, ni d'un majorant de cette probabilité, mais seulement d'une *indication* qui va varier selon le niveau de sûreté atteint par le système. Les indicateurs quantitatifs ne suppriment pas non plus le besoin d'une analyse de sûreté quantitative détaillée du système lors de son implémentation. Ils restent malgré cela utiles pour confirmer des ordres de grandeur et ce que disent les autres indicateurs.

B. Mise en œuvre de la synthèse des coupes

L'algorithme de synthèse est implémenté dans la partie « cœur traitement » de SYNTHESIS. Les données d'entrée incluent le fichier XML des coupes, plus d'autres données fournies par l'utilisateur :

- Liste des coupes minimales fonctionnelles (fichier XML).
- Pour chaque défaillance opérateur apparaissant dans cette liste, type et composant englobant.
- Taux de défaillance associés aux défaillances composants physiques typées, durée de la mission, option minimales seules vs. minimales plus non minimales.

Les données de sortie sont les suivantes :

- Liste des coupes globales avec pour chacune la liste des défaillances composants associées (indicateurs qualitatifs synthétiques).
- Pour chaque coupe globale, liste des coupes minimales fonctionnelles qui lui sont sous-jacentes.
- Pour chaque coupe globale, et pour l'événement redouté, les indicateurs quantitatifs synthétiques associés.

C. *Illustration de la synthèse sur un exemple simple*

Supposons que le composant C_1 contienne les défaillances fonctionnelles $\{f_{11}, f_{12}, f_{13}, f_{14}, l_{15}\}$ (f_{ij} étant une défaillance de type *F* sur C_i). Soient aussi les composants C_2, C_3 et C_4 qui contiennent $\{f_{21}, f_{22}, l_{23}, f_{24}\}$, $\{f_{31}, f_{32}, f_{33}, l_{34}\}$ et $\{l_{41}, f_{42}, f_{43}, f_{44}\}$ respectivement. Supposons enfin qu'il y ait quatre coupes minimales :

$MFC_1 = \{f_{12}, f_{13}, f_{14}, f_{24}, f_{32}\}$ $\quad MFC_2 = \{f_{11}, f_{14}, f_{33}, l_{41}\}$

$MFC_3 = \{f_{31}, f_{32}, l_{41}\}$ $\quad MFC_4 = \{f_{32}, f_{42}\}$

Les transformées par synthèse des coupes minimales fonctionnelles MFC_1, MFC_2, MFC_3, MFC_4 sont les coupes globales suivantes, où F_k et L_l désignent des défaillances composants :

$MGC_1 = \{F_1, F_2, F_3\}$ $\quad GC_2 = \{F_1, F_3, L_4\}$

$MGC_3 = \{F_3, L_4\}$ $\quad MGC_4 = \{F_3, F_4\}$

Parmi ces coupes globales (indicateurs synthétiques qualitatifs), MGC_1, MGC_3 et MGC_4 sont aussi minimales. GC_2 n'est pas minimale. Pour une durée de mission de 2 heures et des valeurs typiques des taux de défaillance, les indicateurs quantitatifs associés aux coupes globales sont les suivants :

IQ(*MGC1*) = (2*5E-6)*(2*5E-6)*(2*5E-6) = 1E-15 ;

IQ(*MGC3*) = (2*5E-6)*(2*4.5E-5) = 9E-10 ;

IQ(*MGC4*) = 1E-10.

Et l'indicateur quantitatif associé à l'ER est :

IQ(ER) = 1E-15 + 9E-10 + 1E-10 = 1E-9.

D. *La traçabilité entre niveaux*

La synthèse consiste en fait en un retour (abstraction) du niveau détaillé « opérateurs fonctionnels » vers le niveau « composants », qu'elle enrichit de nouvelles données tout en assurant la traçabilité entre niveaux. Les défaillances « pertes totales » d'un composant physique correspondent aux défaillances fonctionnelles de type *L* du composant modèle (il peut n'y en avoir qu'une seule). Les « autres dysfonctionnements » du composant physique correspondent aux défaillances fonctionnelles de type *F* comprises dans leur composant modèle. Chaque coupe globale correspond elle-même à l'ensemble des coupes fonctionnelles qui lui sont sous-jacentes. Les indicateurs synthétiques quantitatifs sont des informations additionnelles qui se rapportent au niveau « composants ».

VII. APPLICATION DE SYNTHESIS AU SYSTEME EXEMPLE

On appliquera une première fois le cycle « construction (ou mise à jour) du modèle — évaluation des performances de sûreté ». On constatera que les objectifs de sûreté ne sont pas atteints. Après analyse des indicateurs, l'expert définira une barrière de sûreté supplémentaire. Le cycle à deux étapes mentionné ci-dessus sera alors réitéré. Il faudra en tout trois cycles complets pour atteindre l'objectif.

On notera que les opérateurs, les défaillances fonctionnelles et leurs coupes, bien qu'éloignés des structures physiques réelles des composants, donnent des indications précieuses sur les lacunes des barrières de sûreté existantes, et aident à les combler.

A. *Évaluation initiale de la sûreté*

La durée de la mission est de une heure, avec des taux de défaillance typiques Λ_L = 4.5E-5 et Λ_F = 5E-6. La partie gauche de la Table 1 (cf. § IV.D) montre les résultats du premier cycle (coupes fonctionnelles, coupes globales, indicateurs quantitatifs) pour la version initiale du modèle système. Les coupes globales sont détaillées sur la Table 2. On note la présence de quatre coupes globales minimales d'ordre 1 et de type *F* :

{*FMCOM1*}, {*FMCOM2*}, {*MCDU1*}, {*MCDU2*}.

Les coupes sous-jacentes à ces coupes globales sont les plus critiques. On voit aussi sur les indicateurs quantitatifs que l'ordre de grandeur visé de 1E-7 est loin d'être tenu, en cohérence avec les indicateurs qualitatifs. Il est possible d'analyser ces coupes sous-jacentes et de comprendre comment elles produisent l'ER. L'expert se basera alors sur ces analyses pour définir des barrières de sûreté supplémentaires, ciblées sur ces coupes critiques.

B. *Analyse des coupes fonctionnelles sous-jacentes aux coupes globales d'ordre 1*

1) *Coupes globales {MCDU1} et {MCDU2}*

La coupe globale {*MCDU1*} ne comporte qu'une seule coupe sous-jacente {*MCDU1.EO*}. La sortie de *MCDU1* est corrompue, ce signal (état *KO*) traverse dans *FMCOM1* les opérateurs *FIFO*, *PV*, *TRAJ*, *PATH* (cf. Fig. 2, sortie *FMCOM1_Angles*), puis se propage à travers *FGCOM1* jusqu'à la sortie *FlightDirectors1*, elle aussi corrompue.

Le signal *FMCOM1_FIFO* est injecté en entrée du *PV* de *FMCOM2*. Les deux CRC générés par les *PV* des deux voies étant identiques, il n'y a pas d'alarme CRC sur *FMCOM1*. Par ailleurs la « trajectoire simplifiée » *FMCOM1_TrajSimpl* est corrompue et injectée en entrée de *PATH* dans *FMMON1*. Les signaux *FMCOM1_Angles* et *FMMON1_Angles* sont corrompus (hypothèse pessimiste : identiquement corrompus). Les *Check_Angles* de *FMCOM1* et de *FMMON1* ne lèvent donc pas d'alarme.

On est bien dans un cas d'ER puisqu'au final *FlightDirectors1* est corrompu sans alarme sur la voie 1. De manière analogue, la seule coupe sous-jacente à {*MCDU2*} est {*MCDU2.EO*}. On peut vérifier que {*MCDU2.EO*} produit effectivement l'ER.

2) *Coupes globales {FMCOM1} et {FMCOM2}*

L'outil nous indique que les coupes minimales sous-jacentes à {*FMCOM1*} sont {*FIFO.EO*} et {*PATH.EO*, *TRAJ.TrajSimpl.EO*}, plus deux autres coupes sous-jacentes similaires d'ordre 2. La première coupe {*FIFO.EO*} aboutit à l'ER de la même façon qu'y aboutit {*MCDU1.EO*}. Quant à la deuxième, le signal *FMCOM1_Angles* est corrompu et se propage jusqu'à *FlightDirectors1*. La « trajectoire simplifiée » *FMCOM1_TrajSimpl* corrompue est toujours appliquée en entrée de *PATH* dans *FMMON1* (Fig. 3). La sortie *FMMON1_Angles* est donc corrompue et les monitorings entre *FMCOM1* et *FMMON1* comparent deux informations fausses (hypothèse pessimiste : identiquement fausses). Cette deuxième coupe aboutit donc à l'ER puisque aucune alarme de la voie 1 n'est levée.

De manière analogue, l'analyse des deux coupes sous-jacentes restantes de {*FMCOM1*} et de l'unique coupe sous-jacente {*TX2.EO*} de {*FMCOM2*}, permet de vérifier la présence de l'ER pour ces trois coupes.

Le problème consiste alors à faire évoluer le modèle système de manière à supprimer toutes ces coupes sous-jacentes critiques, i.e. les coupes globales d'ordre 1. Les indicateurs synthétiques refléteront cette évolution.

TABLE 2. COUPES GLOBALES DU MODELE INITIAL

	ADIRS1	ADIRS2	ADIRS3	FGCOM1	FGMON1	FMCOM1	FMCOM2	MCDU1	MCDU2	IQ
1						F				5,0E-06
2							F			5,0E-06
3								F		5,0E-06
4									F	5,0E-06
5	F	F								2,5E-11
6	F	L								2,3E-10
7	F		F							2,5E-11
8	F		L							2,3E-10
9	F			F						2,5E-11
10	L		F							2,3E-10
11		F	F							2,5E-11
12			F	F						2,5E-11
13				F	F					2,5E-11
14				F	L					2,3E-10
										2,0E-05

C. *Ajout d'un « Check_Pil » avant validation du plan de vol par l'opérateur PV*

On suppose que dans une première phase de vol exempte d'ER un « plan de vol » non corrompu a été chargé dans *PV*. Le scénario que nous envisageons maintenant est une mise à jour par le pilote de ce plan de vol.

Le flux de données présentées en entrée de *PV* sur *FMCOM1* constitue en fait un « brouillon » du plan de vol (Fig. 2). Après un délai d'inactivité en entrée de *PV*, ce brouillon est figé et se propage vers la sortie de *PV*, i.e. l'entrée de *TRAJ*. Le monitoring supplémentaire consiste à faire valider par le pilote l'intégrité du brouillon avant de le charger dans *PV*. Les données du brouillon sur la voie 1 sont ainsi renvoyées au pilote (signal en pointillés *FMCOM1_DemConf*) pour confirmation via *MCDU1*. Le pilote compare le brouillon avec les données « plan de vol » qu'il souhaitait charger (*Check_Pil*, opérateur du composant *Pilot*). S'il y a cohérence, il renvoie via *MCDU1* un signal de validation en direction de *PV* (signal en pointillés *MCDU1_Conf*). Le brouillon est alors validé et apparait non corrompu en entrée de *TRAJ*. Dans le cas inverse, la sortie de *PV* vers *TRAJ* reste inchangée, donc non corrompue. Le même mécanisme existe sur la voie 2 via *MCDU2*.

Ce monitoring supplémentaire a pour effet de supprimer les coupes sous-jacentes {*MCDU1.EO*}, {*MCDU2.EO*} et {*FMCOM1.FIFO.EO*}, plus l'unique coupe sous-jacente de {*FMCOM2*}. Les coupes globales {*MCDU1*}, {*MCDU2*} et {*FMCOM2*} disparaissent donc (partie médiane de la Table 1, cf. § IV.D)). Il s'agit bien d'un cas où une évolution du modèle supprime plusieurs coupes globales avec toutes leurs coupes sous-jacentes.

Mais il nous reste la coupe globale {*FMCOM1*} d'ordre 1 et ses trois coupes sous-jacentes, notamment {*PATH.EO, TRAJ.TrajSimpl.EO*}. D'où la nécessité d'un deuxième monitoring supplémentaire.

TABLE 3. COUPES GLOBALES DU MODELE OPTIMISE

	ADIRS1	ADIRS2	ADIRS3	FGCOM1	FGMON1	FMCOM1	FMCOM2	FMMON1	MCDU1	MCDU2	IQ
1	F	F									2,5E-11
2	F	L									2,3E-10
3	F		F								2,5E-11
4	F		L								2,3E-10
5	F			F							2,5E-11
6	F					F					2,5E-11
7	L		F								2,3E-10
8		F	F								2,5E-11
9			F	F							2,5E-11
10			F			F					2,5E-11
11				F	F						2,5E-11
12				F	L						2,3E-10
13					F	F					2,5E-11
14						F	F				2,5E-11
15						F		F			2,5E-11
16						F		L			2,3E-10
17						F				F	2,5E-11
18							F		F		2,5E-11
19									F	F	2,5E-11
											1,5E-9

D. *Ajout d'un « Check_Traj » dans FMMON1 et FMMON2*

Rappelons que l'opérateur *TRAJ* de *FMCOM1* génère une trajectoire en entrée de *PATH* et une « trajectoire simplifiée » *FMCOM1_TrajSimpl* en direction du *PATH* de *FMMON1* qui génère à son tour *FMMON1_Angles*.

Le monitoring supplémentaire consiste à comparer en interne de *FMMON1* les trajectoires simplifiées issues de *FMCOM1* et de *FMCOM2* (Fig. 3, *Check_Traj* en pointillés). Ceci a pour effet de supprimer la coupe sous-jacente {*PATH.EO, TRAJ.TrajSimpl_EO*}. La corruption de la trajectoire simplifiée issue de *FMCOM1* est en effet détectée par cette comparaison, puisque la voie 2 est exempte de pannes. Une alarme voie 1 est générée vers le pilote, nous ne sommes donc plus dans un cas d'ER.

On constate de manière analogue que les deux coupes sous-jacentes restantes de {*FMCOM1*} disparaissent elles aussi. La coupe globale {*FMCOM1*} est donc supprimée. Il s'agit d'un exemple supplémentaire où une évolution du modèle système entraîne la suppression simultanée de plusieurs coupes similaires sous-jacentes à une même coupe globale. La Table 1 (§ IV.D), partie de droite, et la Table 3 illustrent ces résultats. L'indicateur quantitatif valant 1.5E-9, suggère que le niveau de sûreté 1E-7 visé est atteint (à confirmer lors de l'analyse quantitative détaillée du système lors de son implémentation).

VIII. CONCLUSION ET PERSPECTIVES

L'objectif général de l'article était de décrire la méthode outillée SYNTHESIS d'évaluation et d'optimisation de performances de sûreté des systèmes complexes, notamment

des systèmes avioniques embarqués. Les § IV, § V et § VI contiennent cette description générale. Le deuxième objectif de l'article était de décrire l'application de la démarche SYNTHESIS à notre exemple. C'est le thème des § III, § IV et § VII. Enfin, le troisième objectif était de montrer sur notre exemple comment les fonctionnalités du système peuvent évoluer et être optimisées du point de vue de la sûreté à l'aide des indicateurs. Le § VII est consacré à ce point.

À l'issue de la démarche on peut légitimement se demander si on n'obtiendrait pas les mêmes coupes globales sans décomposer les composants en opérateurs, mais simplement en leur associant deux défaillances basiques : *perte totale* (toutes les sorties perdues) et *corruption totale* (toutes les sorties erronées). On générerait ensuite les coupes composants sans passer par les coupes minimales fonctionnelles ni par leur synthèse.

Un essai a été réalisé sur un autre modèle. Il s'avère que les coupes globales obtenues par synthèse contiennent toutes les coupes obtenues en utilisant *perte totale* et *corruption totale*, plus d'autres absentes de cette liste. La différence tient au fait que *corruption totale* ne couvre pas les cas où *une* des sorties du composant est *individuellement* corrompue, ni la combinatoire des modes communs potentiels entre défaillances de sorties.

On peut aussi penser que les coupes minimales fonctionnelles dépendent de la décomposition retenue pour les composants en opérateurs. On pourrait par exemple fondre deux ou trois opérateurs en un seul, ou au contraire en scinder un en plusieurs. L'expérience prouve que les coupes globales générées ensuite par synthèse sont peu sensibles à ces variations. On trouve seulement des écarts quand la logique dysfonctionnelle proprement dite des opérateurs varie. C'est le cas du voteur mentionné au § IV.C pour lequel plusieurs choix sont possibles quant à la logique de vote.

Il faut enfin noter que la portée de l'algorithme de synthèse dépasse le simple contexte du CSA. Il est en effet applicable à n'importe quel jeu de coupes ou séquences, minimales ou non minimales, y compris celles qu'on peut extraire d'une simulation ou d'un arbre de défaillances construit manuellement. Il suffit de spécifier, pour chaque défaillance apparaissant dans les coupes, le « groupe » (i.e. le composant) auquel elle appartient. Des « coupes groupes », minimales ou non minimales, peuvent ensuite être générées.

L'outil SYNTHESIS comprend d'autres fonctionnalités, en fait déjà implémentées, en plus de celles décrites dans cet article, qui n'ont pas encore été complètement validées sur des systèmes exemples. Ces fonctionnalités feront l'objet de futurs articles. Pour la version actuelle, on suppose l'absence de défaillances composants dormantes en début de mission. Pour s'affranchir de cette limite, des temps de réparation variables selon la fréquence des tests (CBIT, PBIT, etc.) peuvent être associés aux défaillances fonctionnelles. De manière analogue, la notion de type peut se généraliser sans limitation sur le nombre. La version complète de la synthèse est capable de traiter ces nouveaux aspects, chaque défaillance composant d'une coupe globale étant caractérisée par son type et son test associés.

En outre, un nouvel algorithme de calcul des indicateurs quantitatifs a été spécifié et implémenté. Il se fonde sur des moyennes de temps d'exposition de défaillances composants sur un ensemble de missions, et non sur les valeurs maximales de ces temps (périodes des tests). Il tient également compte des phases respectives de ces tests cycliques. Ces moyennes sont moins majorantes que les valeurs maximales mentionnées dans cet article.

REFERENCES

[1] Åkerlund O., P. Bieber *et al*, 2006, ISAAC, "A Framework for Integrated Safety Analysis of Functional, Geometrical and Human Aspects", in 3rd European Congress on Embedded Real Time Systems (ERTS), Toulouse, France

[2] Lisagor O, McDermid JA, Pumfrey DJ. "Towards a practicable process for automated safety analysis". In proceedings of the 16th international ship and offshore structures conference (ISSC'06); 2006

[3] Papadopoulos, Y., Walker, M., Parker, D., Rüde, E., Hamann, R., Uhlig, A., Gratz, U., Lien, R. (2011) "Engineering failure analysis and design optimisation with HiP-HOPS". Engineering Failure Analysis 18 (pp. 590-608)

[4] Papadopoulos Y, McDermid JA, Sasse R, Heiner G. "Analysis and synthesis of the behavior of complex programmable electronic systems in conditions of failure". Journal of Reliability Engineering and System Safety 2001; 71(3):229–247

[5] Fenelon P., J. McDermid, July 1993, "An Integrated Toolset for Software Safety Analysis, Journal of Systems and Software, p. 279-290

[6] Arnold A., G. Point, A. Griffault, A. Rauzy, 2000, The AltaRica Formalism for Describing Concurrent Systems, in Fundamenta Informaticae, Vol. 34, p. 109-124

[7] Batteux, M., Prosvirnova, T., Rauzy, A., Kloul, L. (2013). "The AltaRica 3.0 project for model-based safety assessment", In 11th IEEE International Conference on Industrial Informatics (pp. 741–746)

[8] A. Leblond. **«** Synthèse de coupes minimales fonctionnelles en coupes minimales composant », *in Actes du congrès LambdaMu19 (actes électroniques)*. Dijon (France). October, 2014

[9] Marco Bozzano, Adolfo Villafiorita "Improving System Reliability via Model Checking the SAP/NuSMV-SA Safety Analysis Platform" Computer Safety, Reliability, and Security Volume 2788 of the series Lecture Notes in Computer Science pp 49-62

[10] Bozzano M., A. Villafiorita, *et al*, 2003, "ESACS, an Integrated Methodology for Design and Safety Analysis of Complex Systems", in European Safety and Reliability Conference (ESREL), Maastricht. Balkema Publishers

[11] David Parker (University of Hull, UK), Martin Walker (University of Hull, UK) and Yiannis Papadopoulos (University of Hull, UK) "Model-Based Functional Safety Analysis and Architecture Optimisation" Embedded Computing Systems: Applications, Optimization, and Advanced Design