Fiabilité de mission d'un avion. évaluation stochastique en opération

Kossi Tiassou, Karama Kanoun, Mohamed Kaâniche, Christel Seguin, Chris Papadopoulos

Fiabilité de mission d'un avion — Evaluation stochastique en opération

Kossi Tiassou[1], Karama Kanoun [1], Mohamed Kaâniche[1], Christel Seguin[2], Chris Papadopoulos[3]

1. CNRS, LAAS, Université de Toulouse
7 Avenue du Colonel Roche, BP 54200, F-31031 Toulouse, France
prenom.nom@laas.fr

2. ONERA/DCSD/CD,
2 Avenue Edouard Belin, 31055 Toulouse Cedex 4, France
christel.seguin@onera.fr

3. AIRBUS Operations Ltd,
New Filton House, Golf Course Lane, Filton, Bristol, BS99 7AR, United Kingdom,
Chris.Papadopoulos@Airbus.com

RÉSUMÉ. Cet article traite de l'évaluation de la fiabilité de mission des avions en opération pour aider à la planification des missions et de la maintenance. Nous développons une approche de modélisation basée sur un méta-modèle permettant de structurer les informations nécessaires à la construction d'un modèle stochastique. Ce dernier peut être mis à jour en opération afin de tenir compte de la situation courante et d'évaluer la fiabilité de mission en opération. Une étude de cas, relative à un sous-système avion, est considérée à titre d'illustration en utilisant les réseaux d'activités stochastiques.

ABSTRACT. This paper addresses aircraft mission reliability assessment during operation to support aircraft mission and maintenance planning. We develop a modeling approach, based on a meta-model, that is used to structure the information needed to build a stochastic model. The latter can be tuned in operation to take into account the current situation in order to assess the aircraft mission reliability in operation. A case study, related to an aircraft subsystem, is considered for illustration purpose, using the Stochastic Activity Networks.

MOTS-CLES : fiabilité opérationnelle, évaluation basée sur les modèles stochastiques, réévaluation de sûreté de fonctionnement, planification de mission et maintenance.

KEYWORDS: operational reliability, stochastic model-based assessment, dependability re-assessment, aircraft mission and maintenance planning.

1. Introduction

Le transport aérien est devenu un moyen de déplacement usuel pour un nombre de plus en plus grand de personnes. Dans ce contexte, en plus de la garantie d'un niveau de sécurité élevé aux passagers, il est devenu essentiel pour les avionneurs et les compagnies aériennes d'optimiser l'exploitation des avions et leur maintenance. Ainsi, des travaux doivent être menés pour développer des moyens efficaces permettant, en particulier, de prévenir et limiter les risques d'interruptions des vols et éviter ainsi les pertes économiques dues à l'impossibilité d'opérer et à l'insatisfaction des clients.

Notre travail vise à développer une méthode de modélisation de la sûreté de fonctionnement permettant d'évaluer l'aptitude de l'avion à continuer sa mission jusqu'à une destination donnée, où tout éventuel problème pourrait être réglé facilement. Le modèle est destiné à être utilisé durant la planification d'une mission (constituée d'un ensemble de vols et d'opérations de maintenance) et durant sa réalisation. Avant une mission, il sera utilisé pour estimer la durée pendant laquelle l'avion pourra effectuer sa mission sans se retrouver dans un état indésirable. Cette évaluation vise à fournir un support pour l'attribution d'un profil de mission adapté à l'avion (d'autres critères opérationnels qui ne sont pas abordés dans cet article doivent être également pris en compte). Une fois qu'une mission est assignée à l'avion, le modèle sera utilisé en cours d'opération pour évaluer l'aptitude à poursuivre la mission, en cas d'occurrence d'un événement majeur. Le modèle peut ainsi aider à la planification de la maintenance grâce à la comparaison des fiabilités opérationnelles correspondant à différentes solutions de maintenance.

Le but ultime de notre travail est de développer un modèle stochastique de sûreté de fonctionnement qui peut être configuré facilement au cours de la mission pour évaluer la fiabilité opérationnelle de l'avion, tenant compte de l'état opérationnel de l'avion par rapport à la mission en cours ou à effectuer. En opération, la configuration du modèle nécessite une bonne connaissance de l'architecture et du comportement du système, mais ne devrait pas nécessiter la connaissance des techniques de modélisation sous-jacentes. Le modèle est à configurer par des opérateurs qui ont été formés à cet effet. En revanche, la construction du modèle exige à la fois une connaissance approfondie des techniques de modélisation de sûreté de fonctionnement et de l'architecture détaillée ainsi que du comportement du système vis-à-vis des fautes et dans des conditions diverses. Ainsi, le modèle stochastique ne peut être construit que par des spécialistes de la modélisation, avec l'aide de l'avionneur.

La modélisation peut être basée sur les chaînes de Markov, les réseaux de Petri stochastiques et leurs extensions, construits directement ou à partir de la transformation d'un modèle dans un langage dédié, les réseaux d'activités stochastiques, SAN (Movaghar et Meyer,1984), ou le langage AltaRica (Arnold *et al.*, 1999; Boiteau *et al.*, 2006). La seule exigence est que, une fois le modèle construit, il puisse être facilement ajusté de l'extérieur et traité efficacement. Pour l'utilisation finale de notre approche, le modèle sera en AltaRica, langage déjà expérimenté par Airbus. Le module de traitement du modèle est propriétaire et est en

cours de développement. Pour obtenir des résultats nous permettant de vérifier la validité de la démarche, avant la fin du développement de tous les modules propriétaires, nous avons utilisé le formalisme SAN, et son outil associé Möbius (Daly *et al.*, 2000).

Lors de la construction du modèle stochastique, une attention particulière doit être accordée à sa validation. En effet, il n'est pas possible d'effectuer une validation complète du modèle en opération, à cause des contraintes de temps (à l'exception de quelques contrôles simples, tels que des contrôles de cohérence des données modifiées). Il est donc primordial de construire le modèle en suivant un processus bien défini, accompagné d'un processus de validation approfondie. Dans notre approche, nous recommandons de développer d'abord un méta-modèle pour structurer les informations nécessaires à sa construction. D'autre part, compte tenu du fait que ce type d'évaluation sera effectué pour plusieurs familles d'avions, le méta-modèle fournit en plus une approche unifiée pour aider à la construction de modèles correspondant à différents avions. De plus, le méta-modèle permet aussi de mettre en évidence les paramètres mis à jour lors de l'opération de chaque avion. Enfin, le méta-modèle a été pensé pour permettre à terme la compilation d'un même modèle dans différents formalismes comme par exemple les réseaux d'activités stochastiques (SAN) ou AltaRica.

L'article est structuré comme suit. La section 2 donne quelques travaux connexes. La section 3 présente le contexte de modélisation et d'évaluation. La section 4 présente les principes généraux du méta-modèle. La section 5 propose une étude de cas et donne une base pour la construction d'un modèle correspondant, en utilisant le méta-modèle. Une mise en œuvre concrète est donnée à la section 6, en utilisant le formalisme des réseaux d'activités stochastiques (SAN). La section 7 présente des exemples de résultats d'évaluation. Enfin, la section 8 conclut l'article.

2. Travaux connexes

A notre connaissance, la modélisation de la fiabilité opérationnelle des avions a été peu abordée dans la littérature. Les études réalisées sont plutôt concentrées sur les aspects de sécurité (voir (Kehren *et al.*, 2004; Prescott et Andrews, 2005; Ramesh *et al.*, 2008), par exemple), et la plupart des travaux publiés sur la fiabilité opérationnelle ont été réalisés à des fins d'amélioration de la conception des avions (Bineid et Fielding, 2003; Saintis *et al.*, 2009). Dans (Sachon et Paté-Cornell, 2000), la problématique de l'estimation des retards et de la sécurité des vols en rapport avec les activités de maintenance a été abordée. Un modèle probabiliste d'analyse de risque a été développé afin de quantifier l'effet de la politique de maintenance sur l'opérabilité des avions.

Une approche d'aide à la décision, concernant la planification des activités de maintenance, est présentée dans (Papakostas *et al.*, 2010). Les auteurs proposent une méthode pour planifier la maintenance en tenant compte des critères d'optimisation comme le coût, la durée de vie résiduelle et les risques opérationnels. Le travail

effectué ne concerne pas l'évaluation de fiabilité, mais utilise plutôt la mesure de fiabilité comme entrée.

Dans (Ahmadi et Soderholm, 2008), les conséquences opérationnelles des défaillances système ont été analysées en utilisant les arbres d'événements. L'article analyse les éventuelles conséquences des défaillances en tenant compte de la phase de vol durant laquelle elles ont eu lieu. Une approche de modélisation basée sur l'arbre de défaillance du système avion est présentée dans (Saintis *et al.*, 2009), en considérant seulement les événements liés à l'autorisation de vol, sans aborder les conséquences des défaillances en vol.

En ce qui concerne les aspects de modélisation, le problème est généralement catégorisé, comme problème de modélisation de systèmes à missions structurées en phases (PMS – Phased-Mission System). L'approche PMS a été utilisée dans (Meyer *et al.*, 1980) pour analyser les performances d'un calculateur de commande de vol tolérant aux fautes. Le modèle développé consiste en trois niveaux hiérarchiques. L'étude s'est limitée à un seul vol, et la maintenance n'est pas considérée. (Mura et Bondavalli, 2001) analyse les PMS et présente une approche de modélisation de la sûreté de fonctionnement. Il a été démontré que, sous certaines conditions, le modèle peut être traité en utilisant une méthode analytique. (Chew *et al.*, 2008) aborde le problème dans le cadre de la mise en œuvre du concept de MFOP (Maintenance-Free Operating Periods), qui caractérise des périodes durant lesquelles le système est utilisé sans possibilités de maintenance. Le modèle développé est résolu par simulation.

De tous ces travaux, aucun ne vise directement la modélisation de la fiabilité des avions en phase opérationnelle. Les travaux les plus proches (Ahmadi et Soderholm, 2008; Saintis *et al.*, 2009) sont effectués pour une analyse globale de la fiabilité opérationnelle, portant sur toute la période de vie du système. Ces travaux se sont basés sur des arbres de fautes et des arbres d'événements pour l'évaluation de la fiabilité opérationnelle des avions, à des fins de planification de maintenance quand l'avion est en service, en utilisant les modèles stochastiques à espace d'états.

Concernant l'évaluation de la sûreté de fonctionnement en ligne, (Malek, 2008) et (Masci *et al.*, 2011) abordent l'utilisation de l'évaluation en ligne pour faire les choix de solutions les plus adaptées à l'opération d'un système.

3. Contexte de modélisation et d'évaluation

Cette section présente les principaux besoins et contraintes opérationnelles qui doivent être pris en compte lors de la construction des modèles de sûreté de fonctionnement adaptés à notre problématique. Il s'agit de définir les domaines de données à modéliser ainsi que la structuration des données optimisant la mise à jour du modèle.

La réalisation d'une mission est telle que chaque vol est suivi d'une escale où l'avion est préparé pour le prochain vol. A chaque escale, les anomalies observées

lors du vol précédent sont examinées et l'avion est inspecté. Si une défaillance est détectée, une décision doit être prise quant à l'aptitude à effectuer le prochain vol. Le commandant de bord et les agents de maintenance se réfèrent à un document (appelé *Minimum Equipment List*) où les composants sont répertoriés avec le statut Go, Goif ou Nogo.

Le statut Go correspond au cas où l'avion peut voler avec le composant défaillant.

Le statut Goif autorise le vol à condition qu'un certain nombre d'autres équipements soient opérationnels et que certaines procédures opérationnelles ou de maintenance soient possibles. On distingue deux cas :

- Goif-o : Des procédures opérationnelles doivent être effectuées ou réalisables pour pouvoir effectuer le vol. Il s'agit essentiellement d'une limitation des fonctionnalités disponibles pendant le vol.
- Goif-m : Des dispositions concernant la maintenance doivent être prises dans un délai acceptable prédéfini pour régler le problème.

Le statut Nogo empêche l'avion de voler. Dans ce cas, la défaillance doit être réparée avant tout vol. Le vol est autorisé s'il n'y a pas de composants ayant un statut Nogo et si toutes les conditions Goif sont réalisables. Lorsque l'avion ne répond pas aux exigences suite à une défaillance, des activités de maintenance sont entreprises pour résoudre le problème. L'ampleur des conséquences d'une défaillance dépend ainsi de l'aptitude à résoudre le problème avant l'heure de décollage prévue. En fait, le vol n'est considéré comme retardé qu'après le dépassement d'une marge donnée de retard.

Lorsque le vol est autorisé, le vol commence par un roulage vers la piste de décollage. Au cours de cette période, et même après le décollage, le vol peut être interrompu à la suite d'une défaillance critique. L'avion retourne alors à l'aéroport de départ. En fait, le vol peut être dérouté à tout moment si la capacité de l'avion à continuer est dégradée. Les procédures décrites dans les documents comme le FCOM (*Flight Crew Operating Manual*) ou le QRH (*Quick Reference Handbook*) servent de support pour une décision de déroutement (Ahmadi *et al.*, 2010).

Les situations indésirables pendant la réalisation d'une mission sont les *interruptions* de vol, à savoir les *retards*, les *annulations*, les *demi-tours* (retours à l'aéroport de départ) et les *déroutements*.

Les procédures existantes concernent le vol en cours et le prochain vol uniquement. Nos travaux visent à les compléter par des évaluations stochastiques afin de couvrir la totalité de la mission.

3.1. Exigences opérationnelles et mesures à évaluer

Certaines exigences doivent être satisfaites tout au long de la mission. Elles doivent être vérifiées systématiquement lors de l'inspection de l'avion et pendant le vol suite à l'occurrence d'un imprévu tel que la défaillance d'un composant. Si ces exigences ne sont pas satisfaites, le vol n'est pas autorisé. Nous supposons qu'elles

sont satisfaites au moment où l'évaluation est effectuée (date initiale de l'évaluation). Nos objectifs sont d'évaluer i) pendant combien de temps elles resteront satisfaites et/ou ii) la probabilité qu'elles soient satisfaites jusqu'à la fin de la mission planifiée. Si cette probabilité est inférieure à un seuil acceptable donné, une action doit être entreprise.

Nous distinguons deux catégories d'exigences liées au système et à la mission :

- *Min_Sys_Req,* exigences minimales relatives au système, indépendantes des profils de mission, qui doivent être satisfaites quel que soit le vol à effectuer. Elles sont données par la *Minimum Equipment List* et les dossiers de certification des systèmes.
- *M_Prof_Req*, exigences qui sont spécifiques au profil de la mission. Elles sont composées des exigences spécifiques aux vols de la mission.

Nous avons défini deux mesures de fiabilité associées à ces exigences :

- La fiabilité du système (SR) : évaluée par rapport à Min_Sys_Req. Elle est utilisée lors de la planification d'une mission, afin de déterminer le nombre maximum d'heures de vol pouvant être effectuées sans maintenance (ou de manière équivalente, au cours de laquelle les exigences seront satisfaites sans aucune action de maintenance). Elle peut donc être utilisée pour aider à déterminer la durée maximale de la mission.

- La fiabilité de la mission (MR) qui correspond à la probabilité de réaliser la mission avec succès, i.e., les exigences seront satisfaites tout au long de la mission (ou, en d'autres termes, la mission sera accomplie sans interruption opérationnelle). MR est évaluée en considérant Min_Sys_Req et M_Prof_Req. Elle est à évaluer au cours de la réalisation d'une mission, suite à l'occurrence d'un évènement majeur, pour déterminer si une action préventive doit être entreprise, et à quel moment.

Il convient de mentionner qu'afin d'obtenir rapidement des résultats dans certaines situations, on peut envisager d'analyser seulement la fiabilité d'un sous-ensemble de fonctions, identifiées comme critiques pour la mission.

3.2. Les principales informations nécessaires à la construction du modèle

L'évaluation des deux mesures de sûreté de fonctionnement SR et MR est généralement effectuée à partir de modèles stochastiques de sûreté de fonctionnement. Dans notre cas, les modèles doivent compiler des connaissances précises sur l'état d'opération de l'avion. Comme indiqué dans la Figure 1, les informations requises par la construction du modèle peuvent être structurées en quatre catégories : les informations relatives au comportement du système, au profil de la mission, aux exigences et à la maintenance.

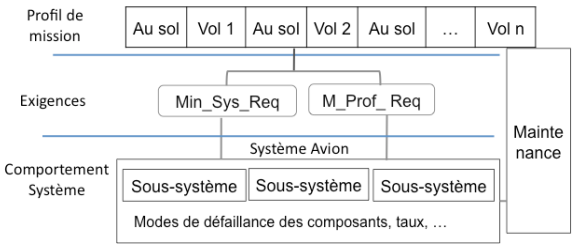

Figure 1 Catégories d'informations

Profil de mission : Il est composé d'informations relatives à la succession des périodes au cours desquelles l'avion est soit en vol, soit au sol (où la maintenance peut avoir lieu), ainsi qu'au nombre et à la durée des vols inclus dans une mission.

Comportement du système : Le système à bord est composé de sous-systèmes et de composants atomiques qui, selon leur état, peuvent fournir ou pas différentes fonctions importantes. La description des scénarios de défaillance et de maintenance des composants, ainsi que les éventuelles pertes de fonctions engendrées constituent un point fondamental dans la construction du modèle.

Exigences : Il s'agit de la représentation de Min_Sys_Req et de M_Prof_Req, formulées comme des compléments d'expressions booléennes, représentant les différentes combinaisons d'états pouvant conduire à une interruption.

Maintenance : Les possibilités de maintenance (en termes de ressources et de durées) sont différentes d'un aéroport à un autre.

3.3. Compétences requises pour construire et mettre à jour le modèle

Traditionnellement, l'analyse de sûreté de fonctionnement à partir de modèles stochastiques est réalisée de bout en bout par un spécialiste en modélisation. Ce spécialiste construit le modèle et le configure pour analyser les différents scénarios d'exécution jugés pertinents. Le modèle est validé par une équipe spécialisée, avec l'aide de l'avionneur, qui est la seule entité à bien connaître le système.

Dans le cadre de notre travail, le scénario d'exécution est déterminé par la situation courante en opération, et la présence de l'équipe spécialisée qui réalise traditionnellement les modèles stochastiques n'est pas garantie en opération. La mise à jour du modèle doit pouvoir être effectuée par des opérateurs formés à cette fin, qui n'ont pas nécessairement une culture de modélisation stochastique, ni une connaissance approfondie de la dynamique de fonctionnement interne du système. Ils feront la mise à jour en se basant sur des procédures préétablies et grâce à une interface conviviale. La Figure 2 résume les étapes du processus de construction et de configuration du modèle.

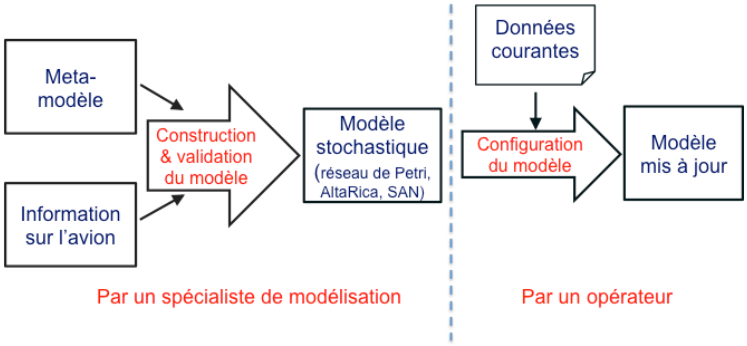

Figure 2 Processus de construction et de mise à jour du modèle

Différents types de changements peuvent avoir lieu lors de la réalisation d'une mission et induire la mise à jour le modèle stochastique en opération. Ces changements peuvent être liés : 1) à l'état des composants du système, 2) aux distributions de probabilité caractérisant les défaillances des composants, 3) au profil de mission (i.e., le nombre de vols ou leur durée), ou 4) aux possibilités de maintenance, ainsi qu'au temps moyen de maintenance des composants défaillants ou la distribution de probabilité des temps de réparation.

La figure 3 donne un aperçu de la mise en œuvre pratique de l'évaluation en opération.

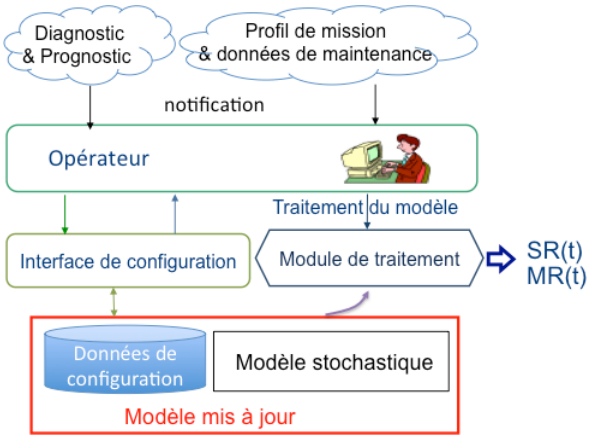

Figure 3 Mise en œuvre de la modélisation en opération

L'ensemble de nos travaux est détaillé dans (Tiassou, 2013). Cet article aborde principalement la construction du modèle stochastique (dont une première version a été publiée dans (Tiassou *et al.*, 2012)), en mettant l'accent sur le méta-modèle. Il présente également quelques modèles en SAN qui sont plus détaillés dans (Tiassou, 2013-b) et résume les principaux résultats du cas d'étude.

3.4. Synthèse de l'impact des besoins opérationnels sur la modélisation

Pour conclure cette section, nous noterons que les modèles adaptés à notre problème sont des modèles stochastiques de sûreté de fonctionnement qui compilent les caractéristiques des systèmes et de mission définies dans la section 3.2. Cette dernière propose aussi un premier niveau de structuration de ces données reflétant des natures d'objets et des champs d'expertise différents : systèmes de l'avion, mission, maintenance, ...

La section 3.3 met en évidence que des niveaux d'expertise élevés et diversifiés peuvent être mobilisés lors de la construction du modèle mais pas lors de sa mise à jour quotidienne. Afin de permettre une mise à jour facile et rapide en opération, il est proposé de décomposer le modèle stochastique en deux parties : 1) un modèle du système dédié à la représentation du comportement du système et des exigences minimales, et 2) un modèle de la mission représentant les informations liées au profil de la mission et qui peut être modifié sans affecter la possibilité d'utiliser le modèle système pour une évaluation de fiabilité. Ainsi, la structure du modèle système est indépendante de la mission pour un avion donné.

Examinons à présent comment ces choix peuvent être formalisés à l'aide de méta-modèles de données, qui pourront aussi ultérieurement être utilisés pour produire des interfaces masquant les détails des modèles stochastiques superflus pour les opérateurs de mise à jour.

4. Le méta modèle

Nous avons choisi de modéliser les concepts structurant la construction des modèles stochastiques pertinents dans notre contexte applicatif à l'aide d'Ecore, afin de bénéficier de l'environnement de développement associé. Après un rappel de cette notation de méta-modélisation, nous présentons plus particulièrement les méta-modèles des parties « système » et « mission » qui soulèvent des difficultés différentes. Les méta-modèles des exigences sont discutés en lien avec chacune de ces deux parties. Le méta-modèle « maintenance » a été réalisé en suivant la même philosophie que le méta-modèle « mission » et ne sera présenté que brièvement.

4.1. Notations du méta-modèle

Le méta-modèle est développé en utilisant les éléments de méta modélisation définis dans EMF (Eclipse Modeling Framework) (Griffin, 2003), plus précisément les notations de Ecore. Ecore est principalement basé sur les concepts des diagrammes de classes UML (Unified Modeling Language). La Figure 4 présente les principaux concepts.

- EClass : représente une catégorie d'éléments essentiels de modèle, caractérisée par des attributs et des opérations, respectivement dénommés EAttribute et EOperation.

- EReference : représente les relations entre les éléments du modèle. D'un point de vue conception orientée objet, il représente le fait que les objets de l'EClass source ont des attributs qui sont des références à des objets de l'EClass de destination. Dans notre cas, l'orientation indique que l'objet source utilise les informations de l'objet de destination. Une EReference peut être déclarée *containment*, exprimant le fait que l'objet de destination fait complètement partie de l'objet source.
- Héritage : représente une relation entre deux EClass, définissant la source comme un sous-type de la destination.

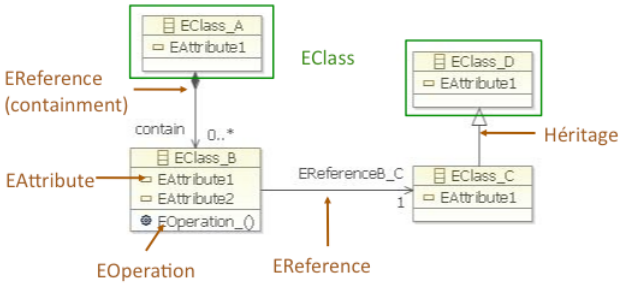

Figure 4 Notations Ecore

4.2. Le méta-modèle du système

Le premier méta-modèle présenté est baptisé méta-modèle « système » car il doit caractériser les principaux constituants des modèles du comportement de l'avion en présence de défaillances et réparation des systèmes embarqués. La difficulté est de déterminer le niveau de granularité et la structure des modèles de propagation de défaillance (et donc du méta-modèle) suffisants pour d'une part faciliter la mise à jour des paramètres des modèles en cours d'opération et pour d'autre part capturer le contenu des modèles de sûreté de fonctionnement utilisés lors des évaluations

La Figure 5 présente le méta-modèle proposé pour répondre à ce besoin et les sections suivantes motivent les choix de méta-modélisation effectués.

4.2.1 Elément facilitant le référencement des éléments du système

Le système à modéliser est organisé en sous-système physiques (par exemple sous-système de commande de vol et sous-système électrique) assurant des fonctions de haut niveau (par exemple génération et distribution d'énergie pour le système électrique). Les sous-systèmes physiques sont eux même constitués de composants physiques élémentaires variés (calculateurs, actionneurs, cables, …).

Les différents paramètres à mettre à jour appartiennent à ces différents éléments. En effet, nous devons mettre à jour les modèles en tenant compte d'observations sur l'état des composants des systèmes physiques et nous devons aussi évaluer l'impact de cet état sur les fonctions à réaliser. Comme indiqué dans la Figure 5, notre méta-

modèle explicite donc les concepts de *Subsystem, SysComponent*, *Function* et demande que chaque instance de ces concepts soit caractérisée par un identifiant *Id* défini pour faciliter le référencement des paramètres mis à jour.

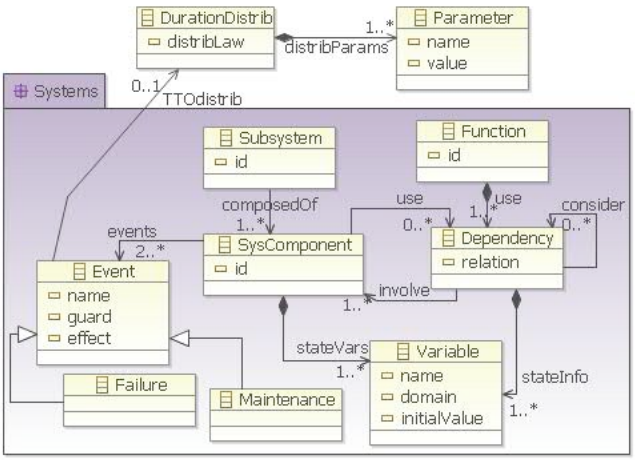

Figure 5 Méta-modèle correspondant au système

4.2.2 Eléments caractérisant le comportement des composants

Les paramètres à mettre à jour durant l'opération dépendent du contenu du modèle de sûreté de fonctionnement permettant d'évaluer les différentes exigences opérationnelles identifiées dans la section 3.1. Les modèles utiles dans notre cas sont des modèles de fiabilité, qui décrivent le comportement du système à l'aide d'états et de transitions comme les processus Markovien ou les réseaux de Petri stochastiques.

Il faut noter que ce type de modèle décrit un espace d'états globaux du système. Or, dans notre cas, les mises à jour doivent être localisées dans les composants et des mécanismes de composition doivent propager l'effet des mises à jour locales sur les états globaux du système, comme dans les formalismes SAN ou AltaRica utilisés dans ce projet pour évaluer les mesures. Nous présentons donc dans cette section les concepts permettant de paramétrer le comportement de chaque composant atomique, nous discuterons dans la section suivante les concepts utilisés pour pouvoir les composer.

Le comportement de chaque composant est fonction de variables d'état et d'événements qui entraînent un changement de valeur des variables d'état. Une variable d'état (EClass *Variable*) possède un nom (*name*), un domaine qui définit les valeurs possibles que peut prendre l'état, et une valeur initiale (*initialValue*) qui correspondra à son initialisation dans le modèle.

Ainsi, le diagnostic d'une défaillance en opération amène à réinitialiser certaines variables d'états

De manière similaire, un événement (EClass *Events* est identifié par son nom (*name*). Les défaillances et les activités de maintenance représentent les principaux événements considérés dans les modèles d'évaluation de la sûreté de fonctionnement. Nous distinguons les événements de type *Failure* et ceux de type *Maintenance* car la définition et la mise à jour des paramètres des événements peuvent être sous la responsabilité d'experts métier différents.

De plus, le calcul des mesures de sûreté de fonctionnement nécessite de caractériser ces événements par des données qualitatives (effets des événements sur le système) et des données quantitatives (intervalles de temps avant défaillances, durées de maintenance).

Ainsi, un événement est caractérisé sur le plan qualitatif par les conditions conditionnant son occurrence (*guard*) et ses effets (*effect*) exprimant le changement dans l'état du système suite à son occurrence.

Ces informations décrivent les lois physiques du système, elles sont a priori constantes.

Leur contenu doit être spécialisé selon le type de modèle stochastique à produire à partir du méta-modèle. Par exemple, les événements correspondent aux « activités » des modèles écrits dans le formalisme SAN présentés en section 6. La valeur de l'attribut « guard » définit pour chaque activité les portes franchies lors de l'exécution de l'activité et les préconditions associées à ces portes. Le méta-modèle proposé n'explicite pas davantage la structure de ces attributs des événements pour deux raisons : ils ne seront a priori pas mis à jour en opération et la liberté laissée permet aux utilisateurs de produire les modèles de sûreté de fonctionnement dans le formalisme de leur choix.

Sur le plan quantitatif, l'événement est défini par la distribution caractérisant le temps jusqu'à son occurrence (*TTOdistrib*). *TTOdistrib* est un objet de *DurationDistrib*, qui vise à représenter la loi de distribution du temps passé dans une situation donnée. La distribution est décrite par le nom de la loi de distribution (*distribLaw* - par exemple, exponentielle ou Weibull) et ses paramètres *distribParams*, qui sont de l'EClass *Parameter*. Un paramètre de distribution est caractérisé par son nom (*name*) et une valeur (*value*).

Le pronostic d'une défaillance à court terme amène à réinitialiser la valeur des paramètres de distribution.

4.2.3 Elément caractérisant les dépendances entre composants

Examinons à présent les concepts de dépendances entre composants d'un même sous-système et entre composants et fonction permettant de produire un modèle stochastique global.

Il peut s'agir de dépendances fonctionnelles induites par exemple par la connexion entre entrées et sorties des composants ; il peut aussi s'agir de relations

plus complexes permettant de modéliser par exemple la propagation de l'état d'un composant sur le reste du système. Concrètement, ces relations de dépendances doivent faire référence à des attributs de composants différents. Cependant, l'accès direct depuis un objet aux attributs d'un autre objet n'est pas toujours possible. La ERelation *Dependency* a été introduite pour pallier ce manque. Elle permet d'une part d'accéder aux valeurs d'attributs de composants distincts. D'autre part, la valeur retournée (*stateInfo*) par l'attribut *relation* permet de spécifier précisément la dépendance existant entre des attributs d'un groupe d'objets.

Les fonctions (*Function*) délivrées par les composants du système sont également décrites en utilisant des instances de *Dependency*, car la valeur de la variable d'état d'une fonction est également déterminée par la combinaison des variables d'état des composantes qui contribuent à sa délivrance.

Ces relations de dépendances sont a priori constantes et doivent être spécialisées selon le formalisme utilisé pour réaliser le modèle stochastique.

4.2.4 Elément caractérisant les exigences applicables aux sous-systèmes

Les fonctions, ainsi que les états des composants système, sont utilisés pour définir les exigences relatives au système. La Figure 6 présente les éléments intervenant dans la définition des exigences.

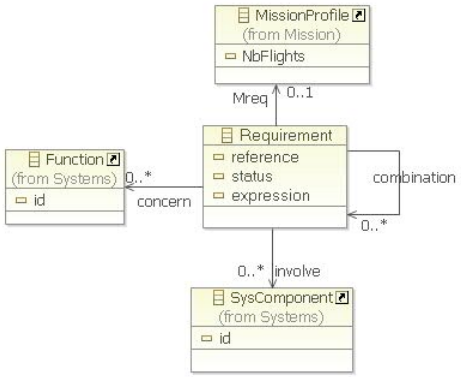

Figure 6 Méta-modèle concernant les exigences

Une exigence est une contrainte liée à une partie du système ou au profil de la mission. Elle doit être satisfaite pour réussir la réalisation d'une partie de la mission. Elle est caractérisée par *(*EClass *Requirement)* : i) l'attribut *reference*, un identificateur qui peut être utilisé pour faire référence à la même exigence dans différentes situations, ii) une variable booléenne *status* qui indique si l'exigence est satisfaite ou non et iii) l'*expression* booléenne qui spécifie la condition à satisfaire par différentes variables d'états des composants ou des fonctions. Enfin, une exigence peut résulter de la combinaison d'autres exigences.

Les exigences qui ne nécessitent pas d'information particulière liée au profil de la mission sont exprimées dans le modèle système.

4.3. Le méta-modèle du profil de la mission

La Figure 7 présente la partie du méta-modèle destinée à la description des informations concernant le profil de mission. Le modèle de sûreté de fonctionnement permettant d'évaluer les conditions de réalisation d'une mission est similaire au modèle utilisé pour évaluer la fiabilité dynamique des systèmes. Néanmoins la description d'une mission est beaucoup plus standardisée que la description des composants d'un sous-système et doit être mise à jour plus régulièrement en opération. Par conséquent, les concepts généraux de variables d'états, d'événements ou de dépendances sont ici spécialisés en tenant compte des concepts « métier ».

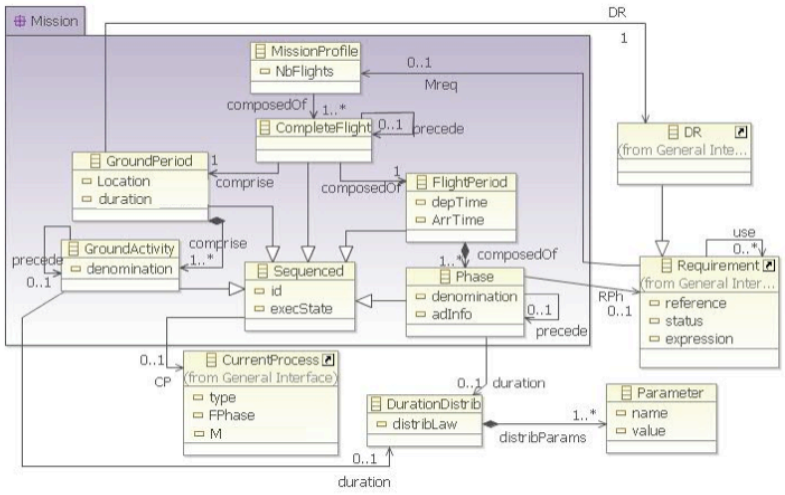

Figure 7 Méta-modèle pour la partie concernant la mission

4.3.1 Elément caractérisant les constituants d'une mission

Un profil de mission est défini par un nombre (*NbFlights*) de vols à effectuer en séquence. Pour la réalisation de chaque vol, une période au sol (*GroundPeriod)* pour préparer le vol, et une période de vol *(FlightPeriod),* qui consiste en l'exécution proprement dite du vol, sont distinguées. L'ensemble du processus pour réaliser le vol est dénommé *CompleteFlight*. La période de vol est décomposée en phases. Une phase est caractérisée par son appellation (*denomination)*, sa durée (*duration)* et des informations additionnelles (AdInfo) qui pourraient être nécessaires dans la définition des exigences.

4.3.2 Elément caractérisant les dépendances entre constituants

Comme le profil de la mission est décomposé en une séquence de périodes et phases, une Eclass *Sequenced* est définie pour représenter leurs caractéristiques communes, qui sont l'identifiant (*id*) et l'attribut *execState*. L'attribut *execState* indique si la partie correspondante de la mission est en cours de réalisation ou pas. L'information sur la partie de la mission en cours de réalisation est transmise au modèle système en utilisant un objet de *CurrentProcess*. L'information concerne le type (période au sol ou phase de vol), l'identifiant de la phase de vol (*FPhase*) s'il s'agit d'une phase de vol et l'information d'autorisation de maintenance si c'est une période au sol. Les objets dérivés de *CurrentProcess* feront partie de l'interface entre le modèle système et le modèle mission.

4.3.2 Elément caractérisant les exigences applicables à la mission

Avant tout vol, l'état opérationnel du système est testé par rapport aux exigences d'autorisation de vol (*DR*), qui représente une synthèse de la Minimum Equipment List. *DR* correspond à *Min_Sys_Req* s'il n'y a pas d'exigences spécifiques à la mission. Les activités de maintenance sont telles qu'elles ne peuvent pas être considérées comme terminées si DR n'est pas satisfait. La réussite de la période au sol est déterminée par la réalisation des activités correspondantes durant la période de temps alloué. La réussite de la période de vol est déterminée par la réussite de ses différentes phases. Une phase est réalisée avec succès si les exigences correspondantes sont satisfaites durant la réalisation.

4.3.3 Elément caractérisant les moyens de maintenance en fonction de la mission

La réalisation des activités de maintenance dépend des moyens logistiques adéquats à l'escale considérée. Une station de maintenance est associée à chaque période au sol et la dépendance est prise en compte en considérant une distribution de probabilités caractérisant le temps nécessaire à la prise en charge des activités correspondantes. Par exemple, une fonction *LDF* (Figure 8) est définie pour prendre en compte le retard dans la prise en charge des activités. *LDF* est une sous-classe de *DurationDistrib* (Figure 5). Des types de stations de maintenance, comme *base principale* et *hors base*, peuvent être utilisés pour les catégoriser. Les tâches sont effectuées en considérant un ordre de priorité (*prioritization*) dans leur réalisation.

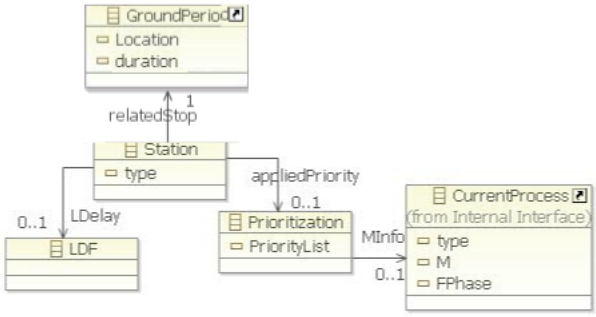

Figure 8 Représentation des informations relatives à la maintenance

4.4. Remarques finales sur le méta-modèle

Il convient de mentionner que le méta-modèle présenté dans cet article est destiné à illustrer la philosophie de l'approche. Les attributs des classes définies peuvent être enrichis avec d'autres informations spécifiques. Seul l'avionneur a une connaissance complète du système et des opérations possibles. Le modèle peut être construit en utilisant un sous-système particulier ou plusieurs sous-systèmes ensemble, surtout s'ils sont fortement dépendants.

Le principal objectif du méta-modèle est d'aider à la construction des modèles stochastiques. Cependant, il peut également être utilisé à des fins différentes. En particulier, il peut être utilisé comme un support pour donner un aperçu du contenu global du modèle. Le méta-modèle peut être aussi utilisé pour la formation des équipes qui seront en charge de la mise à jour du modèle en opération.

5. Cas d'étude

Le cas d'étude concerne un sous-système qui contrôle l'une des surfaces mobiles des avions (Bernard *et al.*, 2007), dénommé CSM dans le reste de l'article. Après avoir décrit brièvement ce système, nous présentons une spécification basée sur les éléments du méta-modèle pour décrire le contenu du modèle stochastique correspondant qui fera l'objet de la partie 5.

5.1. Présentation du système

Le système (Figure 9) est composé de trois calculateurs primaires (P1, P2, P3), d'un calculateur secondaire S1, de trois actionneurs (ServoCtrl_G, ServoCtrl_B et ServoCtrl_Y), d'un module de commande secours (BCM) et de deux composants permettant d'alimenter le module de secours en énergie (BPS_B et BPS_Y).

Les calculateurs sont connectés aux actionneurs, qui font mouvoir la surface. S1 et P1 sont connectés à l'actionneur ServoCtrl_G, P2 est connecté à ServoCtrl_B, et P3 à ServoCtrl_Y. La connexion entre un calculateur et un actionneur forme une ligne de commande qui peut agir sur la surface. On distingue les lignes de commande suivantes :

- PL1: connecte P1 et ServoCtrl_G
- PL2 : connecte P2 et ServoCtrl_B
- PL3 : connecte P3 et ServoCtrl_Y
- SL : connecte S1 et ServoCtrl_G

Il y a aussi la ligne de commande secours BCL qui est formée de BCM, BPS_B, BPS_Y, ServoCtrl_Y et ServoCtrl_B.

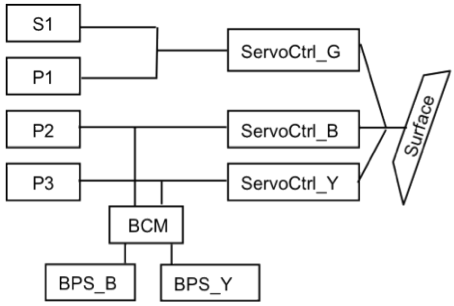

Figure 9 Structure du système

Initialement le calculateur secondaire S1, le module de commande secours BCM, BPS_B et BPS_Y sont inhibés. La surface est alors contrôlée par les trois lignes de commande primaire (PL1, PL2, PL3). Lorsque ces trois lignes de commande primaire sont défaillantes, S1 est activé et SL prend la commande. Si cette dernière défaille aussi, les composants BCM, BPS_B et BPS_Y sont activés pour assurer la commande. Trois modes de contrôle peuvent ainsi être distingués : la commande par les calculateurs primaires (PC), la commande par S1 (SC) et la commande par les équipements de secours (BC). La Figure 10 résume les trois modes de contrôle.

Figure 10 Modes de contrôle et lignes de commande associées

Exigences opérationnelles associées : Selon (MMEL, 2008) :

E1 : la défaillance d'un des composants P2, ServoCtrl_G, ServoCtrl_Y, ServoCtrl_B, BCM, BPS_B ou BPS_Y est un « Nogo ».

E2 : les défaillances de P1, P3 et S1 sont « Goif » :

- P3 et S1 doivent être opérationnels en cas de défaillance de P1.
- P1 et S1 doivent être opérationnels en cas de défaillance de P3.
- P1, P2 et P3 doivent être opérationnels en cas de défaillance de S1.

Il n'existe aucune exigence liée aux profils des missions concernant ce système dans les documents applicables.

5.2. Spécification du modèle système

Le méta-modèle est utilisé pour spécifier le modèle système. Tous les composants système ont des comportements similaires et sont représentés à l'aide des éléments relatifs à *SysComponent* (Figure 5). Le nom de chaque composant est utilisé comme identifiant (*id*). Pour chaque composant x, une variable d'état est considérée, avec un domaine défini par l'ensemble {*ok, failed*}, la valeur initiale est « *ok* ».

Pour les événements à associer, nous considérons un événement de défaillance qui modifie l'état de « *ok* » vers « *failed* », et un événement de maintenance qui restaure l'état à « *ok* ». Nous supposons que l'événement de défaillance se produit en vol. Pour cela, nous utilisons un objet d'interface *CP* (instance de *CurrentProcess* - Figure 7) entre les deux parties du modèle, qui donne des informations sur la partie de la mission en cours d'exécution. Il est aussi utilisé dans l'expression de la garde de l'événement de maintenance pour spécifier l'information d'autorisation de la maintenance (*CP_M*). Par exemple pour le composant P1, les événements sont définis comme suit :

Evénement de défaillance :
name: *P1_failure*
guard: *state=ok and CP-type=flight;* effect *state=failed*
TTOdistrib: *distribLaw=exponential, parameter: lambda=txP1*
Evénement de maintenance :
name: *maintainP1*
guard: *state= failed and id ∈ CP-M;* effect *state=ok*
TTOdistrib: *distribLaw=deterministic*, parameter: *t=1*

Pour le calculateur secondaire et les composants de commande de secours, les scénarios d'activation et de désactivation sont représentés. L'activation et la désactivation dépendent de l'état des lignes de commande primaires. Celles-ci sont représentées en utilisant des instances de *Dependency*. *PL1*, *PL2*, *PL3* représentent respectivement l'état de la connexion entre P1 et ServoCtrl_G, l'état de la connexion entre P2 et ServoCtrl_B, et l'état de la connexion entre P3 et ServoCtrl_Y. Les variables d'état associées (*PL1-stateInfo*, *PL2-stateInfo* et *PL3-stateInfo*) ont l'ensemble {*ok, failed*} comme domaine et leurs valeurs sont déterminées par la fonction de combinaison suivante (*relation*), en utilisant *PL1* comme exemple :

PL1-stateInfo = ok si P1-state=ok et ServoCtrl_G-state=ok

PL1-stateInfo = failed sinon

Il est à noter que cette expression représente juste une formule pour déterminer l'état de la ligne. Dans la pratique, et selon le formalisme choisi pour la construction du modèle, on peut définir directement la valeur de *PL1-stateInfo* dans la spécification des événements qui modifient les états de P1 et ServoCtrl_G.

Les exigences exprimées à la fin du § 5.1 sont relatives aux composants système et sont applicables quel que soit le profil de mission. Elles sont exprimées en tant qu'exigences minimales (Min_Sys_Req), auxquelles peuvent s'ajouter des exigences spécifiques à une mission donnée. Elles sont exprimées en utilisant les éléments définis dans la Figure 6.

L'attribut *reference* n'est pas utilisé ici car il est à utiliser uniquement dans le contexte des mises à jour du modèle. Les exigences sont considérées comme satisfaites initialement (*status=satisfied)*. L'expression de *Min_Sys_R* est formulée comme la conjonction des exigences E1 et E2 (voir § 5.1).

E1: Les exigences opérationnelles liées à « Nogo » sont exprimées comme suit :

$$P2 = ok \wedge ServoCtrl\_G = ok \wedge ServoCtrl\_Y = ok \wedge ServoCtrl\_B = ok \wedge BCM = ok \wedge BPS\_B = ok \wedge BPS\_Y = ok$$

E2: Les exigences opérationnelles liées aux « Goif » sont exprimées comme suit :

- $(P1=ok) \vee (S1=ok \wedge P3=ok)$;
- $(P3=ok) \vee (S1=ok \wedge P1=ok)$;
- $(S1=ok) \vee (P1=ok \wedge P2=ok \wedge P3 =ok)$.

La conjonction de E1 et E2 donne Min_Sys_Req.

$$\begin{aligned} Min\_Sys\_Req = \{ & P2=ok \wedge ServoCtrl\_G =ok \wedge ServoCtrl\_Y =ok \wedge \\ & ServoCtrl\_B=ok \wedge BCM=ok \wedge BPS\_B=ok \wedge BPS\_Y =ok \wedge \\ & (P1 =ok \vee (S1 =ok \wedge P3 =ok)) \wedge \\ & (P3 =ok \vee (S1 =ok \wedge P1 =ok)) \wedge \\ & (S1 =ok \vee (P1 =ok \wedge P2=ok \wedge P3 =ok)) \} \end{aligned} \quad (1)$$

En utilisant les lignes de commande, dont les états sont des combinaisons d'états des composants de base, l'expression devient :

$$\begin{aligned} Min\_Sys\_Req = \{ & PL2 =ok \wedge BCL =ok \wedge \\ & (PL1 =ok \vee (PL3 =ok \wedge SL =ok)) \wedge \\ & (PL3 =ok \vee (PL1 =ok \wedge SL =ok)) \wedge \\ & (SL =ok \vee (PL1 =ok \wedge PL3 =ok)) \}. \end{aligned} \quad (2)$$

5.3 Spécification du modèle mission

Le profil de la mission est spécifié en utilisant les éléments définis dans le §4.2. Nous considérons *p* vols par jour, pendant *d* jours. Il faut créer des instances de *CompleteFlight* correspondant à ces vols. Pour leur identification (*id*), une

numérotation suffit. Pour chaque instance, une période au sol est considérée, avec *gpd* comme durée estimée. La période au sol comprend une période d'activités de maintenance planifiée dont la durée *SM_Time* peut être considérée comme déterministe avec une valeur *smd*. Les activités de maintenance planifiée sont prolongées par des activités de maintenance non planifiée, qui ont généralement lieu lorsque les conditions d'autorisation du vol ne sont pas satisfaites. Les autres activités au cours de la préparation du vol sont considérées comme ayant une durée donnée *oad*.

Les périodes de vol suivant les périodes au sol sont divisées en trois phases dénommées *Taxing_to_Takeoff, In_Flight* et *Landing*, décrites comme suit :

Taxing_to_Takeoff : duration: distribLaw=deterministic, parameter: t=ttd
In_Flight : duration: distribLaw=deterministic, parameter: t=ifd
Landing : duration: distribLaw=deterministic, parameter: t=ld

Les variables *ttd*, *ifd* et *ld* sont les durées estimées pour ces phases.

Nous supposons que les activités de maintenance ont lieu tous les soirs.

6. Le modèle dans le formalisme SAN

Le modèle SAN représente une implémentation exécutable de la spécification précédente, basée sur les éléments du méta-modèle. Une brève description du formalisme SAN est donnée en annexe. La structure du modèle en SAN du cas d'étude est donnée dans la figure 11. Le modèle système est lui même composé de trois modèles correspondant aux trois modes de contrôle PC, SC et BC présentés dans §5.1 et du modèle des exigences système. Dans ce qui suit, nous présentons trois exemples de modèles associés au contrôle primaire (PC), aux exigences système et à la mission.

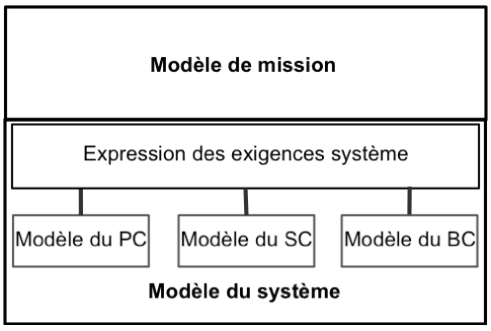

Figure 11 Structure du modèle SAN

6.1 Le modèle SAN du contrôle primaire en SAN

Le modèle SAN du contrôle primaire, PC, est donné à la Figure 12. Les activités *x_failure* représentent les événements de défaillance du composant *x*. Leur activation est conditionnée par la présence d'un jeton dans la place *flight*. Les activités *Maintainx* représentent la maintenance et leur activation est conditionnée par la présence d'un jeton dans la place *CP_M*. La place *flight* (respectivement, *CP_M*) représente le fait que l'avion est en phase de vol (respectivement, en période de maintenance). Leurs marquages sont gérés dans le modèle mission. Pour des raisons de clarté, certaines places impliquées dans les prédicats ou fonctions des portes d'entrée ne sont pas explicitement reliées à ces dernières sur la représentation graphique.

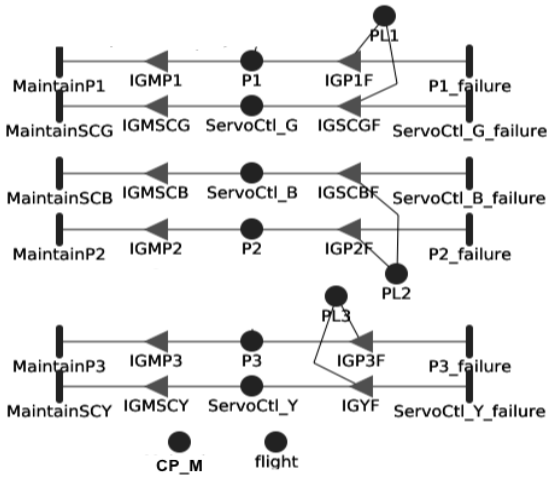

Figure 12 Modèle du contrôle primaire, PC

Les transitions représentant les activités de maintenance (*Maintainx*) sont à gauche des places et les événements défaillance (*x_failure*) à droite. Les portes d'entrée associées contrôlent leurs franchissements. Par exemple, les portes d'entrée *IGP1F* et *IGMP1* sont définies comme suit :

IGP1F Prédicat : P1->Mark() && flight->Mark() Fonction : P1->Mark()=0; PL1->Mark()=0;
IGMP1 Prédicat : P1->Mark()==0 && *CP_M* ->Mark() Fonction : P1->Mark()=1; if(ServoCtrl_G->Mark()) PL1->Mark()=1;

Chaque place $Pli_{i=1,2,3}$ représente l'état de la ligne *PLi*. *PLi* a un jeton quand les places *Pi* et *ServoCtrl_x* correspondantes sont marquées. Les marquages des places *CP_M* et *flight* sont utilisés dans les prédicats des portes d'entrée pour activer les activités de maintenance et de défaillance, comme expliqué plus haut.

6.2 Le modèle des exigences système en SAN

Le modèle de l'expression des exigences constitue l'interface entre le modèle de mission et le modèle système. Il est donné dans la Figure 13. La place *Min_Sys_Req* représente la satisfaction des exigences. Le franchissement des activités instantanées *Fulfilled* et *Not_ Fulfilled* permet de mettre à jour la place en fonction des changements d'état des lignes de commande (satisfaction de l'expression 2 du §5.2).

6.3 Le modèle mission en SAN

Dans cette étude de cas, il est supposé pour des raisons d'illustration que la mission est composée de vols identiques. Le modèle mission est présenté dans la Figure 13, ainsi que sa composition avec le modèle système. Sa partie supérieure modélise les phases d'un vol et sa partie inférieure représente les activités au sol, lors d'une escale.

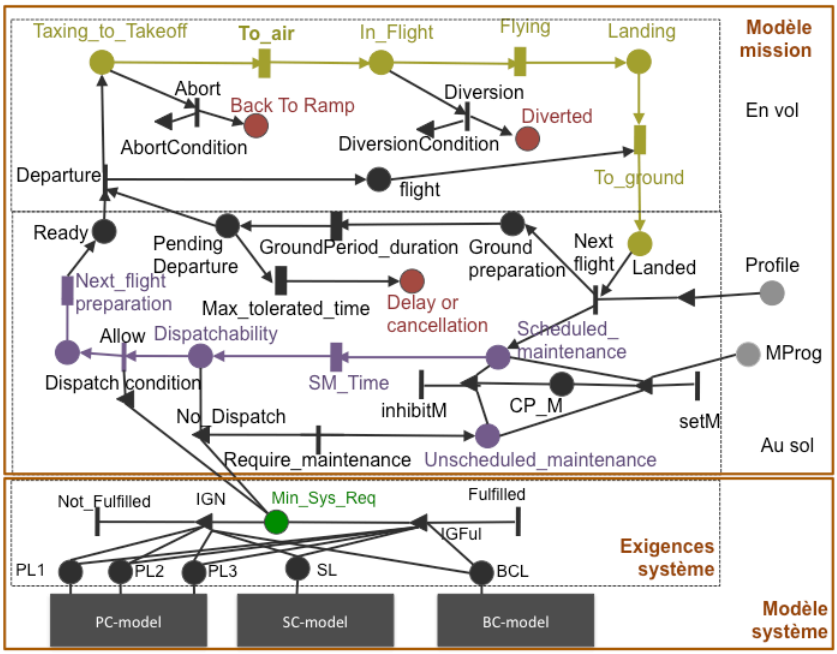

Figure 13 Aperçu du modèle stochastique

Un vol est représenté par trois phases *Taxing_to_Takeoff*, *In_Flight* et Landing. Pendant le décollage *Taxing_to_takeOff* le vol peut être annulé et il peut être dérouté pendant la phase *In_Flight*. Les portes d'entrée *AbortCondition* et *DiversionCondition* définissent les conditions conduisant à l'occurrence de ces interruptions. Le modèle de la partie au sol décrit la préparation pour le prochain

vol. Le début de la préparation est représenté par le marquage des places *Ground_Preparation* et *Scheduled_Maintenance*, indiquant que la phase au sol a commencé avec les tâches de maintenance planifiée (contrôles de routine, par exemple). Lorsque ces tâches sont terminées (franchissement de l'activité *SM_Time*), la place *Dispatchability* est marquée. L'activité instantanée *Allow* peut alors être franchie si les exigences pour autoriser le vol, exprimées dans *Dispatch_Condition*, sont satisfaites. Sinon l'activité instantanée *Require_Maintenance* est tirée car l'action corrective nécessite des tâches de maintenance (indiqué par la condition *No_Dispatch*). La place *Dispatchability* reste marquée jusqu'à ce que l'action corrective réussisse (le prédicat de *Dispatch_Condition* devient vrai) et le vol est autorisé. Dans cette représentation, la satisfaction des conditions pour autoriser le vol consiste à tester le marquage de la place *Min_Sys_Req*. Jusque-là, le temps prévu pour la phase au sol pourrait être dépassé (franchissement de l'activité *GroundPeriod_duration* mettant le jeton dans place *Pending_Departure*) et le délai de retard maximum tolérable (*Max_tolerated_time*) pourrait être dépassé. Un retard ou une annulation peut ainsi advenir. L'activité temporisée *Next_flight_preparation* représente les autres activités (embarquement des passagers, traitement des bagages ...) qui peuvent consommer du temps, occasionnant un retard.

La place *Profile* (à droite) est une place étendue représentant la liste des vols à effectuer. La porte d'entrée reliée à cette place indique s'il y a un prochain vol à effectuer ou non.

La gestion de la maintenance est telle que la place *MProg* représente des listes qui déterminent, pour chaque période au sol, les composants à réparer. La place *CP_M* identifie le composant à réparer et elle est utilisée dans le modèle système pour autoriser la maintenance.

7. Exemples de résultats

Pour traiter le modèle et obtenir les mesures *SR* et *MR*, il faut définir les paramètres du modèle (marquages initiaux des places, les lois de distribution des activités temporisées). Dans le but de préserver la confidentialité industrielle, les valeurs numériques utilisées dans cette partie de l'article ont été sélectionnées pour former un ensemble cohérent, sans correspondre aux valeurs réelles issue de l'observation du système étudié.

Afin d'obtenir des exemples de résultats d'évaluation, nous avons supposé que les événements de défaillance ont des distributions exponentielles. Les taux de défaillance sont supposés entre 10^{-4}/h et 10^{-6}/h.

7.1 Fiabilité système SR

La mesure SR, évaluée à partir du modèle système, correspond à la probabilité que la place *Min_Sys_Req* reste marquée depuis l'instant initial. SR peut aider à l'attribution d'une mission à l'avion, en considérant, par exemple, que sa valeur ne

doit pas être inférieure à un seuil acceptable (appelé exigence minimale de fiabilité, notée MRR). MRR est fixée par la compagnie aérienne, en accord avec l'avionneur. Par souci d'illustration, nous avons considéré MRR = 0,975. La courbe A de la Figure 14 montre la fiabilité du système, obtenue en traitant le modèle système. Elle montre que la durée maximale de mission, sans activités de maintenance, doit être inférieure à 95 heures de vol, pour respecter le seuil MRR = 0,975. Cette évaluation suppose que tous les composants du système sont opérationnels au début de la mission. La courbe B de la même figure suppose que le calculateur P1 est défaillant au début de la mission. Elle montre que, pour satisfaire *MMR*= 0,975, la durée maximale de mission, sans activités de maintenance, doit être inférieure à 45 heures de vol.

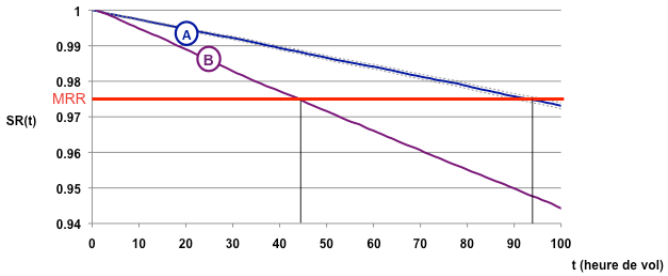

Figure 14 Fiabilité système

7.2 Fiabilité de mission MR

La mesure fiabilité de mission, MR, est évaluée en considérant à la fois *Min_Sys_Req* et *M_Prof_Req*. MR correspond à la probabilité de ne pas avoir de jeton dans les places *Delay Or Cancellation, Back to Ramp* et *Diverted* de la Figure 15. MR est évaluée, en général, au début d'une mission et après l'occurrence d'un événement majeur pour vérifier si une action préventive devrait être entreprise, et à quel moment.

Pour des fins d'illustration, nous considérons une mission typique, d'une durée de 84h, composée de 4 vols identiques par jour sur une semaine. Nous supposons que les activités temporisées du modèle de mission ont des durées déterministes. Il est à noter que différentes distributions peuvent être spécifiées. Chaque vol dure 3 heures. La durée prévue pour une période au sol est de 1,5 heures pendant la journée et 7,5 heures à la fin de la journée (après 4 vols).

Nous examinons d'abord l'impact de la défaillance d'un composant lors de la mission pour montrer comment les résultats de la réévaluation pendant la mission aident à planifier la maintenance du composant. Ensuite, nous montrons l'impact des changements de profil de mission et comment la réévaluation de la fiabilité de mission aidera dans la réaffectation de mission.

7.2.1 Occurrence d'une défaillance de composant

La seule défaillance de P1 ou de S1 n'empêche pas la réalisation d'une mission. Cependant, la défaillance simultanée des deux composants peut conduire à un état global de « Nogo ». La courbe 0 de la Figure 15-a montre la fiabilité de mission, MR, telle qu'évaluée avant la réalisation de la mission, en supposant que tous les composants sont opérationnels au début. On peut voir qu'à la fin de la mission, MR reste supérieure à MRR.

La courbe 0 est la même dans toutes les Figures de 15-a à 15-e, et pour la Figure 16. La courbe 1 de la Figure -b correspond au cas où P1 a été diagnostiqué comme défaillant à la fin du jour 2. MR est réévaluée en considérant i) comme temps initial le jour suivant (i.e., jour 3), et ii) P1 défaillant à l'instant t = 0. On peut voir que MR est inférieure à MRR à partir du jour 5. Ce résultat montre que P1 doit être réparé avant la fin de la mission afin de respecter l'exigence MRR=0,975. Trois cas peuvent être envisagés : P1 est réparé à la fin du jour 3, à la fin du jour 4, ou à la fin du jour 5. Les Figures 15-c et 15-d correspondent respectivement aux cas où P1 est réparé à la fin du jour 3 et le jour 4. On peut constater que dans les deux cas, MR reste supérieure à MRR pour l'ensemble de la mission. Le cas où la réparation a lieu à la fin du jour 5 donne une MR inférieure à MRR, en jour 7. Les résultats ci-dessus montrent que, en cas de défaillance de P1 au jour 2, P1 doit être réparé avant le cinquième jour, en fonction d'où et quand la maintenance peut avoir lieu.

La courbe 4 de la Figure 15-e correspond au cas où P1 a été diagnostiqué comme défaillant à la fin du jour 4. MR est donc réévaluée en considérant i) comme temps initial (t = 0), le jour 5, et ii) P1 défaillant à l'instant t = 0. On peut voir que la nouvelle évaluation est toujours supérieure à MRR à la fin de la mission prévue. La mission peut être poursuivie sans maintenance jusqu'à son terme, à moins qu'un nouvel événement se produise, dans quel cas une nouvelle évaluation est nécessaire.

Les résultats ci-dessus méritent deux commentaires majeurs :

- Nous avons supposé des distributions exponentielles pour tous les évènements de défaillance des composants pour montrer que les changements opérationnels induiront des changements perceptibles dans les résultats. Avec l'approche de modélisation utilisée et les outils disponibles, il est possible d'envisager d'autres distributions et de prendre en compte le vieillissement des composants impliqués dans l'analyse (voir (Tiassou 2013)). Cependant, le vieillissement est un processus très lent. La variation durant un vol et même une mission est généralement très faible, voire nulle, et l'impact de cette variation sur les mesures évaluées n'est pas perceptible.
- MR est égale à 1 au début de chaque nouvelle évaluation, car le système est complètement inspecté après la détection d'une défaillance d'un composant, et il est globalement dans un état opérationnel au moment où la mission est reprise.

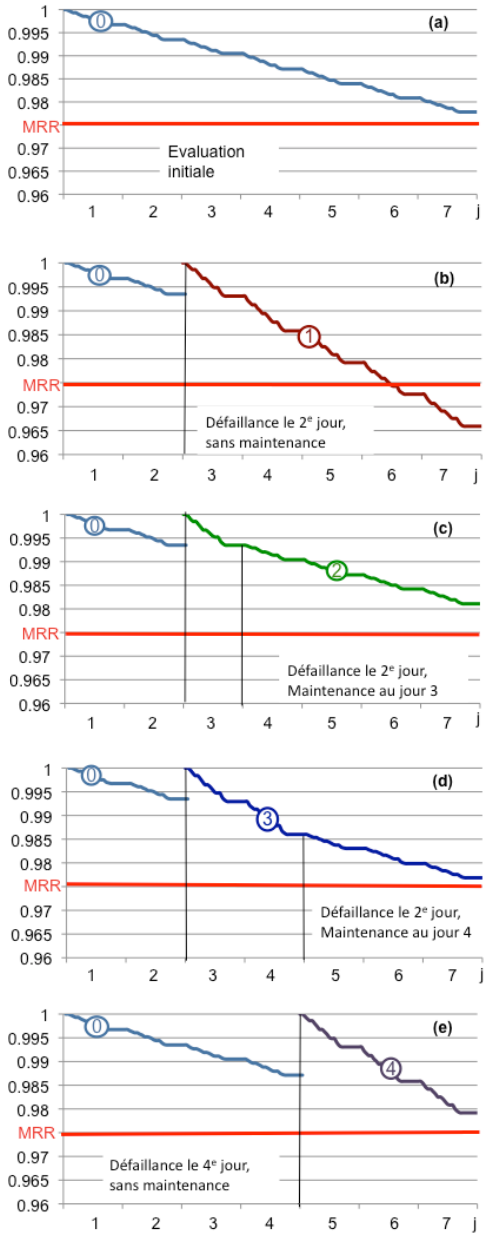

Figure 15 Impact de la défaillance et de la maintenance de P1 sur la fiabilité de la mission

7.2.2 Défaillance du calculateur secondaire S1

Les courbes 5 et 6 de la Figure 16 montrent la réévaluation de MR après la défaillance du calculateur secondaire S1 pendant les jours 2 et 4 respectivement. Ces courbes sont à comparer respectivement aux courbes 1 et 4 de la Figure 15-b et 15-e. La courbe 5 est en dessous de la courbe 1 et la courbe 6 est en dessous de la courbe 4. Cela signifie que S1 a un impact plus négatif sur la fiabilité de mission que P1. Cela est dû au fait que le taux de défaillance de P1 est supérieur à celui de S1. Les exigences sont telles que l'un des calculateurs P1 et S1 doit être opérationnel pour pouvoir réaliser la mission. Par conséquent, le risque d'interrompre la réalisation de la mission est plus élevé lorsque S1 est défaillant que lorsque P1 l'est.

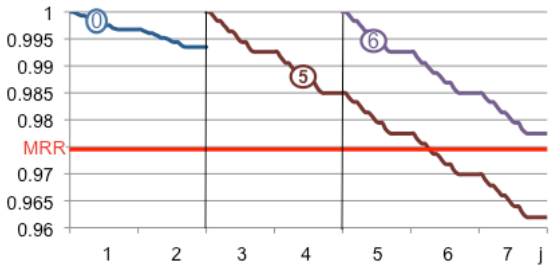

Figure 16 Impact de la défaillance de S1 pendant la mission

L'opération d'un avion dépend de divers facteurs externes. En particulier, certains événements imprévus, qui n'affectent pas forcément directement l'avion lui-même, peuvent entraîner des changements dans la mission initiale. Par exemple, on peut attribuer à un avion de nouveaux vols de durées différentes, ou des vols supplémentaires qui ont été initialement attribués à un autre avion qui doit aller en maintenance. De tels changements nécessitent une réévaluation de la fiabilité de mission. Pour illustrer l'impact des changements dans le profil de la mission, nous avons considéré quatre profils PR0 à PR3, définis à la Figure 17.

PR1 à PR3 supposent des re-planifications de mission à partir du jour 2. La Figure 18 montre que les valeurs de fiabilité pour PR1 sont inférieures à celles de PR0 au bout de 6 jours. Toutefois, l'exigence de fiabilité (MRR = 0,975) est toujours satisfaite. Cela signifie que la nouvelle mission est acceptable. Pour PR2 (Figure 19), MR devient inférieure à MRR. Cela signifie que la nouvelle mission n'est pas acceptable après le jour 6. On peut envisager d'adapter ce nouveau profil afin d'améliorer la fiabilité de mission. Une possibilité d'ajustement pourrait correspondre à PR3. Avec le profil ajusté PR3, MRR est à nouveau satisfaite.

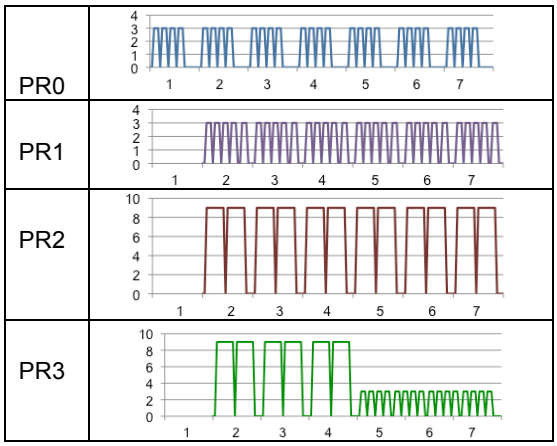

Figure 17 Profil de mission

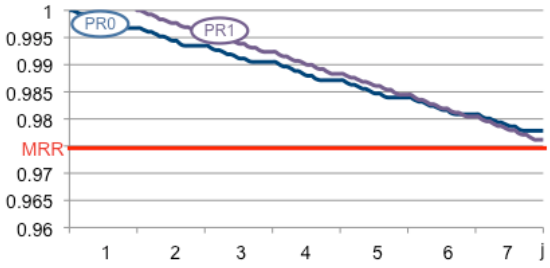

Figure 18 Changement de mission de PR0 à PR1

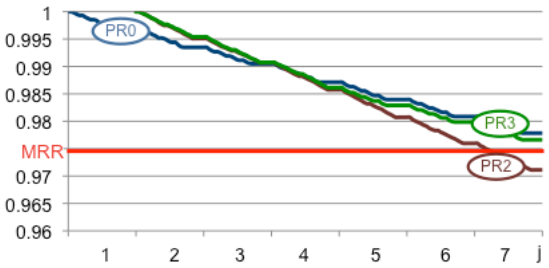

Figure 19 Ajustement de mission de PR2 à PR3

8. Conclusion

Dans cet article, nous avons présenté un modèle qui peut être utilisé pour évaluer la fiabilité opérationnelle des avions avant et durant la réalisation d'une mission de plusieurs jours. Les procédures actuelles d'autorisation de vol concernent le vol en cours et le prochain vol uniquement. Nos travaux visent à les compléter par des évaluations stochastiques, afin de couvrir la totalité d'une mission constituée de plusieurs vols successifs.

Nous avons développé une approche de modélisation permettant de satisfaire cet objectif en tenant compte des spécificités des systèmes avion et de la façon dont les missions sont réalisées. L'approche considère en plus de la construction du modèle de fiabilité opérationnelle, la mise en place d'une base commune pour aider à la construction de modèles correspondant à différentes familles d'avions. Une caractéristique importante qui a guidé le développement de l'approche proposée dans cet article concerne la facilité de mettre à jour le modèle en opération. Pour ce faire, un méta-modèle a été proposé comme support de construction de modèles stochastiques pouvant être facilement mis à jour et traités pour obtenir les résultats d'évaluation. Ce méta-modèle a été utilisé pour construire le modèle correspondant à un sous-système d'avion et pour présenter des exemples de résultats issus du traitement du modèle.

Le cas d'étude illustre l'utilisation du modèle de fiabilité i) pour évaluer la fiabilité opérationnelle d'un avion avant et pendant sa mission, et ii) pour ajuster l'évaluation lorsque des changements significatifs, affectant les états des composants système ou le profil de la mission, se produisent. L'analyse des résultats montre comment l'évaluation peut aider à planifier la maintenance ou à ajuster une mission, suite à l'occurrence des changements durant la réalisation de la mission.

L'évaluation de fiabilité opérationnelle en service pourrait être utile à tout système dont l'opération nécessite une certaine surveillance. Les résultats obtenus représentent donc une base préliminaire qui devrait stimuler le développement de nouveaux moyens, intégrant de plus en plus l'évaluation de la sûreté de fonctionnement dans l'aide à l'exploitation optimale des systèmes.

Annexe : Le formalisme SAN

Les réseaux d'activités stochastiques sont une extension des réseaux de Petri. Le formalisme consiste en quatre objets fondamentaux : les places, les activités, les portes d'entrée et les portes de sortie. Les activités sont l'équivalent des transitions en réseaux de Petri. Elles sont soit temporisées ou instantanées. Les activités temporisées ont une durée et une loi de distribution associée. Les activités instantanées représentent des actions qui s'exécutent immédiatement après activation. Les portes d'entrée sont utilisées pour contrôler l'activation des activités et pour définir les changements de marquage après le franchissement d'une activité. Chaque porte d'entrée est définie avec un prédicat d'activation et une fonction. Les portes de sortie sont comme les portes d'entrée et sont utilisées pour modifier l'état

du système après le franchissement d'une activité. Une porte de sortie est définie seulement avec une fonction. Les fonctions définissent les changements de marquage à appliquer après le franchissement d'une activité. Les portes d'entrée et de sortie sont représentées graphiquement par des triangles (voir Figure 20).

Une activité est activée lorsque les prédicats de toutes les portes d'entrée reliées à l'activité sont vrais, et toutes les places connectées en entrée ont un marquage non nul. Une fois activée, une durée est échantillonnée pour déterminer le temps avant son franchissement. Quand une activité est tirée, l'état du modèle est mis à jour en soustrayant des jetons aux places connectées en entrée vers les places connectées en sortie, et en exécutant les fonctions des portes d'entrée et de sortie.

L'outil Möbius permet la construction de modèles composés. En effet, pour un grand système, il peut être utile de construire le modèle global à partir de sous modèles moins complexes. Ceci est possible en utilisant les opérateurs « Join » et «Replicate». L'opérateur « Join » permet de combiner plusieurs modèles partageant un certain nombre de variables d'état. L'opérateur « Replicate » est utilisé pour créer des copies de modèle ; les copies sont combinées en un modèle global. Les copies peuvent avoir des variables d'état partagées.

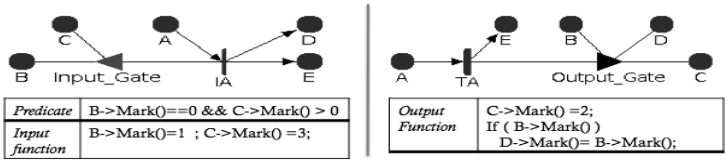

Figure 20 Portes d'entrée et de sortie

Références

Ahmadi A., Kumar U., Söderholm P., 2010. Operational Risk of Aircraft System Failures. International Journal of Performability Engineering, 6, 149 - 158.

Ahmadi A., Söderholm P., 2008. Assessment of Operational Consequences of Aircraft Failures: Using Event Tree Analysis. Actes 2008 IEEE Aerospace Conf., 1-14. Big Sky, MT, USA. Consultable : http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4526622.

Arnold A., Point G., Griffault A., Rauzy Antoine, 1999. The AltaRica formalism for describing concurrent systems. Fundamenta Informaticae, 40, 109–124.

Bernard R., Aubert J., Bieber Pierre, Merlini C., Metge S., 2007. Experiments in model-based safety analysis: flight controls. Consultable : http://sites.google.com/site/pierrebieber/publications/DCDS07_FlightControlsModel_RB.pdf [Consulté le 29 septembre 2010.

Bineid M., Fielding J.P., 2003. Development of a civil aircraft dispatch reliability prediction methodology. Aircraft Engineering and Aerospace Technology, 75, 588-594.

Boiteau M., Dutuit Y., Rauzy A., Signoret J.-P., 2006. The AltaRica Data-Flow Language in Use: Assessment of Production Availability of a MultiStates System. Reliability Engineering and System Safety, 91, 747–755.

Chew S.P., Dunnett S.J., Andrews J.D., 2008. Phased mission modelling of systems with maintenance-free operating periods using simulated Petri nets. Reliability Engineering & System Safety, 93, 980 - 994.

Daly D., Deavours D.D., Doyle J.M., Webster P.G., Sanders W.H., 2000. Möbius: An Extensible Tool for Performance and Dependability Modeling. Actes11th International Conference on Computer Performance Evaluation: Modelling Techniques and Tools, 332–336. TOOLS'00. Londres, BG. Consultable : http://portal.acm.org/citation.cfm?id=647809.737955.

Griffin C., 2003. Introduction to the Eclipse Modeling Framework. Actes MDA™ Implementers' Workshop - Succeeding With Model Driven Systems, Burlingame, Californie, USA. Consultable : http://www.omg.org/news/meetings/workshops/MDA_2003-2_Manual/Tutorial_4_Griffin.pdf

Kehren C. *et al.*, 2004. Advanced simulation capabilities for Multi-systems with Altarica. Actes the 22nd International System Safety Conference, 489-498.

Malek M., 2008. Online Dependability Assessment through Runtime Monitoring and Prediction. Actes Seventh European Dependable Computing Conference, 181-181. Kaunas, Lithuania. Consultable : http://dx.doi.org/10.1109/EDCC-7.2008.27.

Masci P., Martinucci M., Di Giandomenico F., 2011. Towards Automated Dependability Analysis of Dynamically Connected Systems. Actes Tenth International Symposium on Autonomous Decentralized Systems, 139–146. ISADS'11. Washington, DC, USA. Consultable : http://dx.doi.org/10.1109/ISADS.2011.23.

Meyer J.F., Furchtgott D.G., Wu L.T., 1980. Performability Evaluation of the SIFT Computer. IEEE Transactions on Computers, C-29, 501-509.

Movaghar A. Meyer J.F., 1984,.Performability Modeling with Stochastic Activity Networks, *Actes Real-Time Systems Symposium*, Austin, TX.

MMEL, 2008. Master Minimum Equipment List - AIRBUS A-340-200/300. Consultable : http://fsims.faa.gov/wdocs/mmel/a340-200-300 original 05-30-08.pdf [Consulté 25 juin 2014].

Mura I., Bondavalli A., 2001. Markov regenerative stochastic Petri nets to model and evaluate phased mission systems dependability. IEEE Trans. Comput., 50, 1337-1351.

Papakostas N., Papachatzakis P., Xanthakis V., Mourtzis D., Chryssolouris G., 2010. An approach to operational aircraft maintenance planning. Decision Support Systems, 48, 604-612.

Prescott D.R., Andrews J.D., 2005. Aircraft safety modeling for time-limited dispatch. In Proceedings of the Annual Reliability and Maintainability Symposium, 139-145. Alexandria, VA, USA. Consultable : http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1408352.

Ramesh A., Twigg D., Sharma T., 2008. Advanced methodologies for average probability calculation for aerospace systems. Actes 26th international congress of the aeronautical sciences, 1-9. Anchorage - Alaska, USA. Consultable : http://www.icas.org/ICAS_ARCHIVE_CD1998-2010/ICAS2008/PAPERS/332.PDF.

Saintis L., Hugues E., Bes C., Mongeau M., 2009. Computing in-service aircraft reliability. Int. Journal of Reliability, Quality and Safety Engineering, 16, 91-116.

Tiassou K., 2013. Aircraft operational reliability — A Model-based approach and case studies. thèse de doctorat. Univ. Toulouse, INSA, Toulouse, France

Tiassou K., Kanoun K., Kaâniche M., Seguin C., Papadopoulos C., 2012. Impact de l'évaluation de la fiabilité opérationnelle pendant la mission d'un avion, Congrès de Maîtrise des Risques et de sûreté de Fonctionnement (Lambda Mu 17), Tours (France), 16-18 octobre 2012, 9p.

Tiassou K., Kanoun K., Kaâniche M., Seguin C., Papadopoulos C., 2013-b. Aircraft operational reliability — A model-based approach and a case study, RESS (Reliability Engineering and System Safety), 120 (2013) pp. 163–176