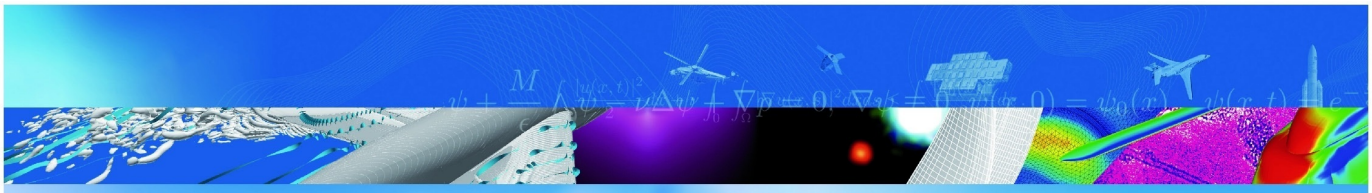
Modèles Dysfonctionnels pour la Gestion de la Qualité de Service des Systèmes Critiques

C. Sannino, J. Sprauel, C. Seguin



COMMUNICATION A CONGRES

Modèles Dysfonctionnels pour la Gestion de la Qualité de Service des Systèmes Critiques

C. Sannino (Thalès), J. Sprauel (Thalès),
C. Seguin (ONERA)

Lamda Mu 20

SAINT-MALO, FRANCE

11-13 octobre 2016

TP 2016-656

70 2016 ans

ONERA

THE FRENCH AEROSPACE LAB

Modèles Dysfonctionnels pour la Gestion de la Qualité de Service des Systèmes Critiques

Dysfunctional Models for Managing the Quality of Service of Safety Critical Systems

Christian Sannino, Jonathan Sprauel
Thales, Toulouse
{prenom.nom}@fr.thalesgroup

Christel Seguin
ONERA, Toulouse
christel.seguin@onera.fr

Résumé

Cet article présente une approche de modélisation et d'analyse permettant d'optimiser la qualité des services rendus par des systèmes qui doivent satisfaire des exigences de sûreté de fonctionnement. Un modèle de Gestion des Modes Dégradés (GMD) est proposé pour faciliter la capture des connaissances sur l'architecture et les modes de fonctionnement d'un système en phase amont de conception. Les spécifications GMD peuvent ensuite être compilées en d'autres modèles mathématiques afin d'assister la conception du système de différentes façons. Une compilation en modèle AltaRica permet de vérifier qu'une spécification satisfait bien les exigences de sécurité qui lui sont applicables. Une compilation en processus décisionnels markovien sous contrainte de chemin permet de concevoir les politiques de reconfiguration de modes qui optimisent la qualité de service du système. Un prototype d'outil a été développé pour faciliter la réalisation et l'analyse des modèles GMD. L'outil a été appliqué à un système avionique afin d'évaluer expérimentalement la validité du concept.

Summary

This paper presents a modelling and analysis approach for optimizing the quality of the services provided by systems under dependability requirements. A model of Management of Degraded Modes (MDM) is introduced to facilitate the capture of knowledge about architecture and system operation modes in early design phase. The MDM specifications can then be compiled in other mathematical models to assist the design of the system in different ways. A compilation into AltaRica models supports the verification of the safety requirements applicable to the system. A compilation into Markov decision processes under path constraint supports the design of reconfiguration policies that optimize the quality of the system services. A prototype tool was developed to facilitate the implementation and analysis of MDM models. The tool has been applied to an avionics system to experimentally assess the validity of the concept.

---

1. Introduction

Cet article présente une approche de modélisation et d'analyse permettant d'optimiser la qualité des services rendus par des systèmes soumis à des contraintes de sûreté de fonctionnement.

En effet, les évolutions de la complexité des systèmes critiques hautement intégrés ou adaptatifs conduisent à proposer de nouveaux modèles formels d'évaluation de la sûreté de fonctionnement. Ainsi, le paradigme du « model based safety assessment » (MBSA) propose des modèles de propagation de défaillance plus compositionnels, dynamiques et proches des modèles de conception que les modèles classiques tels que les arbres de défaillance ou les processus markoviens. Le MBSA est aujourd'hui introduit dans les processus de conception des systèmes critiques pour vérifier efficacement qu'une solution proposée par une équipe de conception satisfait des contraintes de sécurité (cf [1], [2]).

Néanmoins, le retour d'expérience montre tout d'abord que le bénéfice du MBSA pourrait être optimisé en facilitant la capture des modèles aux différentes étapes de conception des systèmes et plus particulièrement en phase amont de conception des systèmes. En effet, lors des phases amont, la connaissance sur le système reste relativement abstraite et il est intéressant de disposer de modèles simples, facilitant la capture de telles connaissances. De plus, si la solution n'est pas satisfaisante, si la spécification du besoin évolue ou si le système doit être utilisé dans de nouvelles conditions, le travail doit être itéré : les concepteurs fournissent une nouvelle solution à modéliser puis évaluer. Cet effort mérite d'être limité non seulement en allégeant l'effort de modélisation mais aussi en proposant des outils qui aident à synthétiser des solutions correctes. Ainsi, des pistes de recherche sont explorées pour, par exemple, allouer correctement aux composants d'un système des taux de fiabilité (cf [3]) ou des niveaux d'assurance de développement (cf [4], [5]). D'autres travaux assistent la synthèse d'éléments de dynamique du système (cf [6]), par exemple pour optimiser la maintenance d'un business jet (cf [7]).

L'enjeu du travail présenté dans cet article est de fournir un cadre de modélisation et de raisonnement intégrant ces différentes avancées dans un processus industriel outillé. Le cadre et les outils développés se proposent d'assister plus efficacement la capture et l'adaptation d'une solution pour satisfaire des contraintes de sécurité et optimiser différents critères. Ce support peut être utilisé à court terme en phase de conception amont du système et à plus long terme en opération, pour adapter le système en temps réel.

Pour satisfaire cet objectif, nous proposons tout d'abord un formalisme de haut niveau permettant de modéliser des systèmes reconfigurables qui doivent satisfaire des exigences de sécurité et fournir différents services avec une qualité minimale. Ce formalisme de « Gestion de Modes Dégradés » (GMD) est centré sur les notions de modes de fonctionnement des ressources du système et de qualité de service produits par les ressources selon leur mode. Son objectif est double : faciliter la capture des connaissances sur le système en phase amont de conception et permettre la traduction vers les modèles mathématiques utilisés pour orienter les choix de conception.

Nous présentons ensuite comment les modèles GMD peuvent être compilés en deux types de modèles mathématiques (modèles AltaRica et processus décisionnel markovien sous contraintes de chemins) ainsi que les types de calculs pouvant être réalisés après traduction : calculs classiques d'évaluation de sécurité réalisables à l'aide d'outils existant ou calculs de stratégies

de reconfiguration du système respectant les contraintes d'optimalité et de sécurité applicables au système, réalisés à l'aide d'algorithmes originaux d'aide à la décision.

Enfin, nous avons prototypé une interface de saisie et d'affichage graphique des modèles GMD, couplé avec différents moteurs de calcul pour tester la validité de la proposition et la faisabilité d'un processus industriel outillé. Nous présentons les résultats obtenus sur un cas d'étude avionique.

2. Modélisation de la Gestion de Modes Dégradés

2.1 Modèle de Gestion de Modes Dégradés

Nous proposons le formalisme de Gestion des Modes Dégradés (GMD) suivant pour modéliser de manière abstraite les systèmes dynamiquement reconfigurables sujets à des défaillances.

Un modèle (GMD) capture les dépendances entre *ressources* d'un système (i.e. les composants physiques ou fonctionnels du système) en fonction de leurs *modes* de fonctionnement nominaux ou dégradés. Les ressources reçoivent et produisent des *services* de *qualité* variable. Les dépendances sont créées en *connectant services fournis et services attendus*. On notera que dans les architectures offrant des services redondants, un service attendu peut être connecté à différents services fournis afin de bénéficier de la meilleure qualité de service.

L'enjeu du modèle est de représenter d'une part la dynamique des modes des ressources et d'autre part l'impact des modes sur la qualité des services du système.

Les ressources ont un *mode de fonctionnement initial* et les modes des ressources pourront évoluer au cours du temps suite à des *transitions*. Ces transitions peuvent être *exogènes* ou *contrôlables*. Les transitions exogènes surviennent sans contrôle de la part du système par exemple lors de l'occurrence d'une défaillance ou d'un changement de l'environnement du système. Leur occurrence peut être caractérisée par une loi probabiliste. Les transitions contrôlables sont initiées par le système en fonction de la situation correspondante, elles sont déterministes et représentent des reconfigurations fonctionnelles ou opérationnelles.

Les transitions sont définies localement pour chaque ressource par leur *effet* et leur *garde*. L'effet d'une transition est l'affectation d'un nouveau mode à ressource concernée. La garde d'une transition est une condition sur le mode de la ressource concernée et sur les qualités de services reçues et fournies par cette ressource. La transition ne peut être franchie que si la condition exprimée dans la garde est satisfaite.

Enfin, on s'intéressera aux *contrats de qualités de service* des ressources. Le contrat d'une ressource fixe d'une part le niveau minimal de qualité requis pour chaque service attendu. Il fixe aussi le niveau de qualité minimal garanti pour chaque service fourni sous réserve que le contrat de qualité attendu soit respecté. Les niveaux attendus et garantis doivent être détaillés pour tous les modes de fonctionnement de la ressource.

Formellement, un modèle GMD est une structure <R, M, S, Q, $\approx_S$, T, QAG> telle que :

- R est un ensemble fini représentant les ressources r du système.
- M est un ensemble fini représentant les modes de fonctionnement ou modes dégradées pouvant être pris par les ressources du système au cours du temps. On note M(r) le sous-ensemble des modes pouvant être pris par une ressource r.
- S est un ensemble fini représentant les services s attendus et produits par les ressources. On note :
  S_in(r) le sous-ensemble des services attendus par une ressource r.
  S_out(r) le sous-ensemble des services fournis par une ressource r.
  Ces ensembles sont disjoints.
- Q est l'ensemble des qualités pouvant être associées aux différents services au cours du temps. On note Q(s) le sous-ensemble des qualités pouvant être associés à un service s ; Q(s) est totalement ordonné.
- $\approx_S \subseteq$ SxS est la relation de connexion entre services fournis et attendus telle que :
  pour tout service s et s' de S, si s $\approx_S$ s' alors il existe r et r' de R tels que s $\in$S_out(r) et s' $\in$S_in(r)
- i :R→M est la fonction qui définit le mode initial de chaque ressource
- T est l'ensemble des transitions de mode pour chacune des ressources du système. Soit t une transition contrôlable ou exogène définie pour une ressource r à l'aide des attributs suivants :
  Garde(t) $\subseteq$ M(r) x Q(S_in(r)) x Q(S_out(r)): la garde caractérise l'ensemble des valeurs de modes et des qualités de services entrant et sortant pour lesquelles la transition peut être franchie
  Effet(t) $\in$ M(r) : (r) : l'effet caractérise le mode atteint après franchissement de la transition
  Taux(t) est un taux de défaillance caractérisant éventuellement les transitions exogènes.
- QAG représente le contrat de qualité de service des ressources selon leurs modes. Il est défini par les deux fonctions suivantes :
  QA : RxMxS → Q définit les contrats de qualité de service attendue. QA(r, m, s) est défini pour toute ressource r, tout mode m de M(r) et tout service s de S_in(r) utilisé par r.
  QG : RxMxS → Q définit les contrats de qualité de service garantie. QG(r, m, s) est défini pour toute ressource r, tout mode m de M(r) et tout service s de S_out(r) fourni par r.

2.2 Exemple de modèle de Gestion de Modes Dégradés

Le concept peut être illustré à partir d'un exemple simplifié inspiré de systèmes de navigation avion.

Une ressource de navigation doit calculer le cap de l'avion à partir de l'estimation de sa position et de sa vitesse. Deux centrales inertielles constituent les ressources utilisées pour estimer la vitesse de manière redondante. La position est estimée à l'aide d'une autre ressource, un GPS, et de l'une des deux centrales inertielles.

On suppose que ces 4 ressources du système peuvent avoir trois modes de fonctionnement. Elles peuvent être dans les modes « erreur » (le service fourni est erroné), KO (le service n'est pas disponible) ou OK (le service est fourni correctement). Initialement les ressources sont OK. On suppose que les transitions ne sont possibles que du mode OK vers le mode KO ou vers le mode « erreur ». Ces transitions sont exogènes, elles représentent l'occurrence de défaillances.

Les ressources « centrale inertielle » et « GPS » fournissent des vitesses ou positions sans attendre de service. Le contrat de qualité garantie fixe que la qualité du paramètre de vol estimé est égale à la valeur du mode de la ressource. La qualité « erreur » est jugée comme étant la pire pour ce type système, le domaine de qualité est donc ordonné de la manière suivante : erreur < KO < OK.

Lorsque la ressource « navigation » est dans le mode OK, elle fournira un cap de qualité OK si elle dispose d'une position et d'une vitesse estimée OK. Dans les autres modes, la qualité du cap est égale à la valeur du mode de la ressource.

L'interconnexion des ressources via les services et représentée dans la figure 1 ci-dessous, qui précise aussi les qualités des services échangés lorsque toutes les ressources sont OK (état initial).

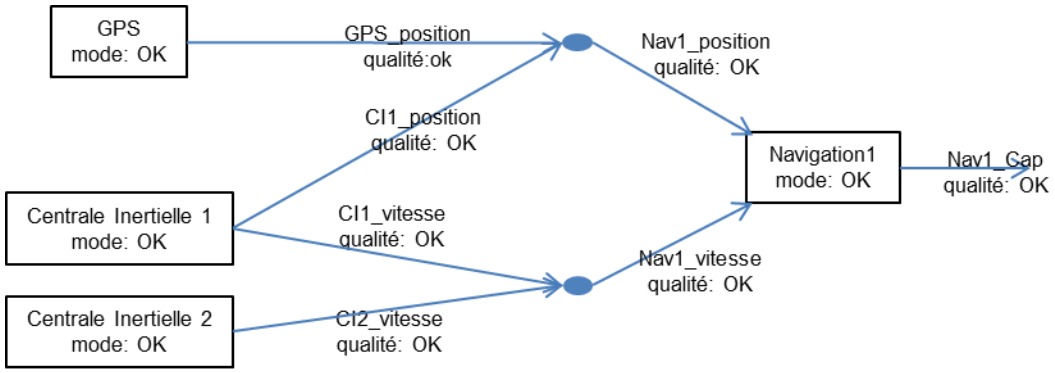


Figure 1: Système de navigation 1 en mode nominal de fonctionnement

Ce système de navigation peut être dupliqué et le choix du cap à utiliser pour guider l'avion est effectué par une ressource de sélection. Cette ressource peut avoir deux modes : PRIM (le cap est calculé par la ressource Navigation 1) ou SEC (le cap est calculé par la ressource Navigation 2).

Initialement, le sélecteur est en mode PRIM. Une transition de mode a lieu lorsque le service Nav1_Cap a une qualité inférieur à OK. Cette transition modélise la détection par le sélecteur d'une qualité Nav1_Cap inadéquate et le basculement du système sur la seconde voie de navigation. Il s'agit d'une transition contrôlable définie pour pouvoir reconfigurer le système.

Le sélecteur attend deux services Nav1_Cap et Nav2_Cap sans contrainte particulière. Le contrat de qualité de service garantie ne dépend que du mode du sélecteur : en mode PRIM, la qualité garantie est celle de Nav1_Cap ; en mode SEC, la qualité garantie est celle de Nav1_Cap.

L'interconnexion du sélecteur avec les ressources de navigation est représentée dans la figure 2 ci-dessous, qui précise aussi les qualités des services échangés pour une configuration de modes où Navigation 1 est KO, Navigation 2 est OK et le sélecteur est en mode SEC.

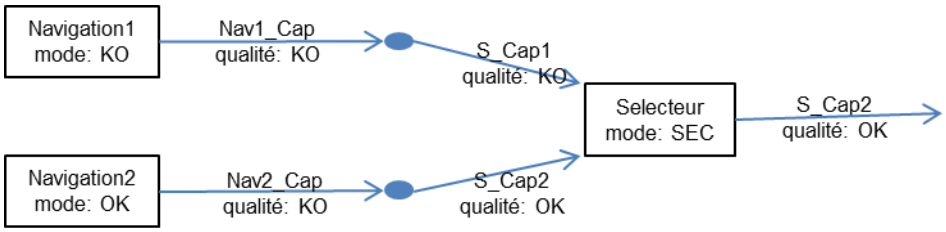


Figure 2: Système de navigation global en mode secondaire

2.3 Dynamique d'un modèle de Gestion de Modes Dégradés

Examinons à présent les règles de comportement d'un modèle GMD, i.e. la façon dont les modes des ressources et les qualités des services sont liés à un instant et comment ils peuvent évoluer au cours du temps.

Soit un modèle GMD <R, M, S, Q, i, T, QAG>. On note m et q les fonctions qui assignent respectivement un mode à une ressource et une qualité à un service à un instant donné:

- m : R→M, tel que m(r) appartient à M(r) pour toute ressource r de R
- q : S→Q, tel que q(s) appartient à Q(s) pour tout service q de Q

m et q définissent le vecteur d'état courant du modèle GMD.

Initialement, m=i.

q est calculée à partir de m, des connexions entre service et des contrats de qualité de service associés à chaque ressource.

Le calcul des qualités des services attendus par une ressource suit les règles suivantes. 1) Si la connexion entre services du système laisse un service attendu déconnecté de tout service fourni, alors ce service attendu déconnecté a la qualité minimale de son domaine de qualités. 2) Si un service attendu est connecté à plusieurs services alors le service attendu hérite de la qualité du meilleur des services reçus. En effet, si un service attendu peut être fourni par différentes ressources avec des qualités différentes, nous privilégierons par défaut l'usage du meilleur service fourni pour optimiser la qualité des services produits à leur tour.

Le calcul des qualités des services fournis par une ressource suit les règles suivantes. 3) Si chaque service attendu par une ressource a une qualité supérieure ou égale à la qualité attendue fixé dans le contrat de la ressource, la ressource fournira ses services avec les qualités garanties. On dit dans ce cas que la ressource est dans un état cohérent avec celui de son environnement. 4) Dans le cas contraire, au moins un service attendu par la ressource n'a pas la qualité attendue, l'état de la ressource et son environnement sont incohérents. Les contrats de qualité des services offerts par la ressource ne sont plus garantis et la ressource fournit des services de qualité minimale.

Ce type d'hypothèses simplifie la capture des connaissances sur le système mais les modèles de fonctionnement résultant peuvent être jugés pessimistes. En effet, lorsque la ressource reçoit des services de qualité supérieure à la qualité attendue, elle peut éventuellement rendre des services de qualité supérieure à la qualité garantie. Réciproquement, lorsque la ressource reçoit des services de qualité inférieure à la qualité attendue, elle peut éventuellement rendre des services dégradés mais de qualité supérieure à la qualité minimale atteignable. Néanmoins, en phase amont de conception, ce choix de modélisation nous semble suffisant car nous souhaitons principalement évaluer si les contrats sont respectés ou pas.

Ces règles de calcul peuvent être formalisées de la manière suivante :

1. Qualité d'un service attendu déconnecté
   $\forall s \in S, ((\forall s' \in S \text{ non}(s' \approx_S s)) \rightarrow q(s)=\text{Min}[Q(s)])$

2. Qualité d'un service attendu connecté
   $\forall s \in S, ((\exists s' \in S \; s' \approx_S s) \rightarrow q(s) = \text{Max}[q(s'')/ \; s'' \in S \text{ et } s''=s])$

3. Qualité des services fournis par une ressource consistante :
   $\forall r \in R, ((\forall s \in S\_in(r), q(s) \geq QA(r, m(r), s) \rightarrow (\forall s' \in S\_out(r), q(s')= QG(r, m(r), s')))$

4. Qualité des services fournis par une ressource inconsistante :
   $\forall r \in R, ((\exists s \in S\_in(r), q(s) < QA(r, m(r),s) \rightarrow (\forall s' \in S\_out(r), q(s')= \text{Min}[Q(s')]))$

Le vecteur d'état défini à partir de m et q évolue lorsqu'une transition de mode d'une ressource est franchie. On dit qu'une transition t d'une ressource r est franchissable lorsque (m(r), q(S_in(r), q(S_out(r)) appartient à Garde(t). Lorsqu'une transition t de la ressource r est franchie, m(r) prend la valeur Effet(t) et m(r') reste inchangé pour tout $r' \neq r$.

[8] propose un algorithme permettant de calculer les valeurs des qualités de l'ensemble des services attendus et fournis conformément à ces règles. Il a été démontré que l'algorithme converge vers une unique solution sous réserve que les contrats de qualité attendue / garantie ne sont pas définis circulairement. Ce résultat vient du fait que la définition des relations entre les services attendus et fournis est particulièrement restrictive et que les valeurs des qualités de service décroissent au cours des itérations du calcul.

3. Analyse des modèles GMD

Les modèles GMD ont été définis pour faciliter la capture des spécifications amont d'un système critique reconfigurable en permettant d'exprimer les dépendances entre ressources et services par des interconnections souples et des contrats de qualité de service simples. Leur définition s'efforce aussi de capturer les informations minimales requises pour être compatible avec des modèles mathématiques préexistants et effectuer sur les modèles des analyses bien fondées. Deux types de compilation ont été effectués : une compilation des modèles GMD en modèle AltaRica permet d'effectuer des évaluations de sécurité classiques sur la spécification du système ; une compilation des modèles GMD en processus décisionnel markovien sous contrainte de chemin permet de générer des tables de reconfiguration optimisant la qualité des services du système.

3.1 Compilation vers les modèles AltaRica et analyse de sécurité

Le langage AltaRica permet de modéliser des dépendances entre composants d'un système à l'aide d'automates de mode (pour capturer la partie « dynamique » de la relation de dépendance) interfacés via des flux d'informations et des assertions (pour capturer la partie « combinatoire booléenne» de la relation de dépendance). Il a été défini pour permettre une modélisation efficace de systèmes et la compilation de modèles AltaRica en modèles classiques de la sûreté de fonctionnement comme les arbres de défaillances ou les processus stochastiques. Plusieurs variantes du langage et les outils de compilation associés ont été définies au cours du temps afin d'améliorer l'efficacité des outils d'analyse associés ou l'expressivité du langage (voir par exemple [9] pour la première version du langage ou [10] pour la plus récente). Le formalisme GMD peut être vu comme une restriction de AltaRica compatible avec l'ensemble de ses variantes. Nous rappelons brièvement les concepts d'AltaRica introduits dans [9] avant de montrer les principes de traductions des modèles GMD en AltaRica.

Les modèles AltaRica sont définis sur des *domaines* de valeurs, i.e des ensembles, pouvant être infinis voire abstraits, mais devant être dénombrables. Les constantes désignent des valeurs particulières des domaines. Les modèles sont structurés en *nœuds* atomiques ou hiérarchisés qui représentent des composants, des sous-systèmes réels ou abstraits. Un nœud a plusieurs paramètres :

- Les variables de *flux* représentent les interfaces du nœud. Elles permettent de le connecter avec d'autres composants (noeuds), comme des câbles pour des systèmes électriques. Certains dérivés du langage Altarica font explicitement la différence entre les variables d'entrée (in) et celles de sortie (out). La valeur d'une variable de flux peut dépendre à la fois de la valeur de la source où elle est connectée et de certaines variables d'état, propres au nœud dans lequel ces entrées/sorties sont définies.
- Les variables d'*état* représentent les états internes du nœud, modes de fonctionnement ou modes opérationnels.
- Les *évènements* permettent d'observer les changements de valeur des variables d'état.
- Les *transitions* définissent les règles de changement des variables d'état. Une transition a une garde (expression booléenne sur les variables de flux et d'état), qui donne une pré-condition pour que la transition puisse être activée. Elle a aussi un effet : le changement d'une ou plusieurs variables d'état. Une transition est associée à un évènement qui permet d'observer le changement d'état.
- Les *assertions* sont des expressions booléennes décrivant des invariants sur les variables de flux et d'état. Elles servent en particulier à spécifier les dépendances entre variables de flux d'entrée et de sortie au sein d'un nœud. Elles permettent aussi d'établir les connexions entre les interfaces de deux nœuds différents.

La traduction des modèles GMD en AltaRica est assez directe. Les ressources de GMD sont des nœuds particuliers de AltaRica, interfacés par des services. Les contrats de qualité de service de GMD peuvent être vu comme des assertions particulières de AltaRica et le concept de transition de GMD est calqué sur celui d'AltaRica. Ainsi, un modèle GMD <R, M, S, Q, i, T, QAG> est traduit en AltaRica de la manière suivante :

1. Pour chaque ressource $r \in R$, on crée un noeud Altarica ayant le même nom *r* tel que :
    - Pour chaque service $s \in S\_in(r) \cup S\_out(r)$ entrant ou sortant de r, on crée une variable de flux du même nom *s* (et du type correspondant in/out en Altarica Dataflow). La variable de flux a pour domaine Q(s), l'ensemble énuméré des qualités possibles pour s.
    - On crée une variable d'état *status* ayant pour domaine M(r), l'ensemble énuméré des modes de la ressource.
    - On crée les transitions appropriées entre les modes à partir de T(r) ; les noms des évènements associés aux transitions sont les noms des modes préfixés par une chaîne de caractère connue.
    - On crée *coherence,* une variable de flux auxiliaire à valeur booléenne vrai/faux
    - On définit l'assertion *coherence= (s1≥QA(r,status,s1) & … si ≥QA(r,status,si))* en fonction des critères de qualité attendue minimale pour chaque service si entrant de r.
    - Les assertions sont créées à partir d'une logique ternaire simple : si *coherence* est vrai, chaque sortie *s* vaut QG(r,status,s) ; sinon chaque sortie vaut la valeur minimale de son domaine.
2. Enfin, un nœud principal est créé pour instancier les nœuds précédents et effectuer la liaison entre les services entrants et sortants.

Les outils associés à AltaRica (voir par exemple http://altarica.labri.fr/ et http://openaltarica.fr/getting-started/) permettent de réaliser sur le modèle traduit différentes analyses de sécurité comme l'extraction d'arbre de défaillance ou la recherche des combinaisons d'événements amenant à un événement redouté.

3.2 Compilation vers les processus décisionnels markovien et optimisation des reconfigurations

Les processus décisionnels markoviens (ou Markov Decision Process MDP) ont été introduits en particulier dans le domaine de la planification (cf [11]) pour résoudre des problèmes d'optimisation de comportement de systèmes dynamiques en environnement probabiliste. Il s'agit de processus markoviens à temps continus ou discrets dont les transitions stochastiques sont annotées par des récompenses. Par exemple, la récompense associée à une action de reconfiguration peut –être la qualité d'un service après avoir effectué cette action.

Plus formellement, un processus décisionnel markovien est une structure (S, A, R, T, I) telle que :

- S est un espace d'états dénombrable
- A est un ensemble dénombrable d'actions
- R : SxA $\rightarrow$ Réels est une fonction de récompense
- T : SxAxS$\rightarrow$ [0 ;1] est une fonction de transition, définissant la probabilité de passer d'un état à un autre par une action
- I : S $\rightarrow$ [0 ;1] est une distribution de probabilité initiale, fixant la probabilité qu'un état soit état initial.

Les algorithmes d'optimisation associés calculent une politique d'actions optimale i.e. le choix des actions à privilégier dans chaque état du processus pour optimiser l'espérance des récompenses cumulées en parcourant les états du processus. Une politique peut-être déterministe (une unique action est choisie dans chaque état) ou stochastique si elle associe à chaque état plusieurs actions selon une distribution de probabilité. Par exemple, lorsque la récompense est la qualité d'un service privilégié pour l'utilisateur, une politique optimale de reconfiguration du système pourra maximiser l'espérance de qualité de service obtenue dans les différents modes de fonctionnement du système.

Néanmoins, les systèmes les plus critiques se doivent aussi d'atteindre les états de fonctionnement les plus dangereux avec des probabilités inférieures à des seuils acceptables. Par exemple en aéronautique, les états de fonctionnement de système ayant des conséquences catastrophiques pour l'aéronef ou ses passagers doivent avoir des probabilités d'occurrence inférieures à 10^{-9} par heure de vol. Les processus décisionnels markovien sous contrainte de chemins ont été introduits pour générer des politiques optimales et satisfaisant les contraintes de sécurité (cf [6]).

Dans ce contexte, la compilation des modèles GMD vers les MDP s'effectue de la manière suivante.

Soient un modèle GMD <R, M, S, Q, i, T, QAG> et m et q les fonctions qui assignent respectivement un mode à une ressource et une qualité à un service à un instant donné comme défini en 2.3. Alors le MDP $<S_{GMD}, A_{GMD}, R_{GMD}, T_{GMD}, I_{GMD}>$ à temps discret associé au modèle GMD est tel que :

- L'espace S_{GMD} des états est le produit M(r1)xM(r2) …xM(rn) des modes de l'ensemble des ressources de R
- I_{GMD} est le vecteur de modes (i(r1), …, i(rn)).
- L'ensemble A_{GMD} des actions est l'ensemble des transitions contrôlables de T
- La fonction de transition T_{GMD} : $S_{GMD}xA_{GMD}xS_{GMD} \rightarrow$ [0 ;1] est définie pour tout triplet (s, a, s') de $S_{GMD}xA_{GMD}xS_{GMD}$ tel que, dans l'espace d'état dérivé du modèle GMD initial, s' est un vecteur d'état atteignable depuis le vecteur d'état s après l'exécution de la transition contrôlable a, suivie d'au plus une transition exogène e.
  Le taux T_{GMD} (s, a, s') vaut alors :

$$\frac{\text{Somme (taux des transitions exogènes menant à s' après l'exécution de a depuis s)}}{\text{Somme (taux des transitions exogènes franchissables après l'exécution de a depuis s)}}$$

- La récompense R_{GMD} : $S_{GMD}xA_{GMD} \rightarrow$ Réels associe à (s,a) le produit suivant :
  Qualité de services préférés q(s_p) dans l'état atteint après avoir effectué l'action depuis s, multipliée par l'espérance du temps moyen passé dans cet état avant une transition exogène.

On notera que l'expression de la récompense nécessite de compléter le modèle GMD avec des critères de préférence sur les services et leurs qualité.

On notera aussi que cette compilation fait alterner transitions exogènes de l'environnement avec une action contrôlable du système en réponse au changement. La compilation alternant transitions exogènes avec une procédure i.e. une séquence d'actions contrôlables insécables a été aussi étudiée pour raisonner sur des reconfigurations de modes non atomiques.

Enfin, les contraintes de chemin sont ajoutées au modèle MDP pour prendre en compte les contraintes de sécurité applicables au système. Elles expriment que le taux d'apparition d'une qualité de service insuffisante doit être inférieur à un seuil acceptable dans le MDP généré.

Des algorithmes originaux ont été proposés dans [6] et [8] pour générer des politiques *optimales* i.e. permettant d'obtenir la meilleure espérance de qualité de service selon les préférences de l'utilisateur à partir de l'état initial et *valides* i.e. respectant l'ensemble des contraintes de chemin applicables au système.

4. Validation expérimentale de l'approche

Examinons à présent comment les modèles GMD et les possibilités d'analyse obtenues après compilation en d'autres formalismes adéquats ont été appliqués chez Thalès pour valider expérimentalement l'approche.

4.1 Le cas de la conception de la fonction RNP-AR

La fonction RNP-AR (Required Navigation Performance - Authorization Required) permet à un avion d'effectuer une approche de haute précision lors de l'atterrissage quand les conditions environnementales et l'avionique utilisable pour réaliser la fonction offrent des garanties suffisantes sur l'intégrité et la précision des données produites (cf [12]). Cette fonction critique a été choisie comme cas d'étude pour étudier la faisabilité et la pertinence d'un processus outillé de conception de système à base de modèles GMD.

Le sous-ensemble de ressources de haut niveau de la figure 3 a été modélisé. Il comprend deux chaînes redondantes de ressources « FMS » offrant des services de gestion du vol : calcul du chemin horizontal (HPATH), calcul des paramètres de guidages (Flight Guidance), …. Ces chaînes utilisent la position et les paramètres de vol de l'avion estimés à l'aide de ressources « GPS » et « IRS ». Les interactions avec les pilotes (saisie de consigne / visualisation de résultat) sont réalisées par les ressources «MCDU » et « PFD ».

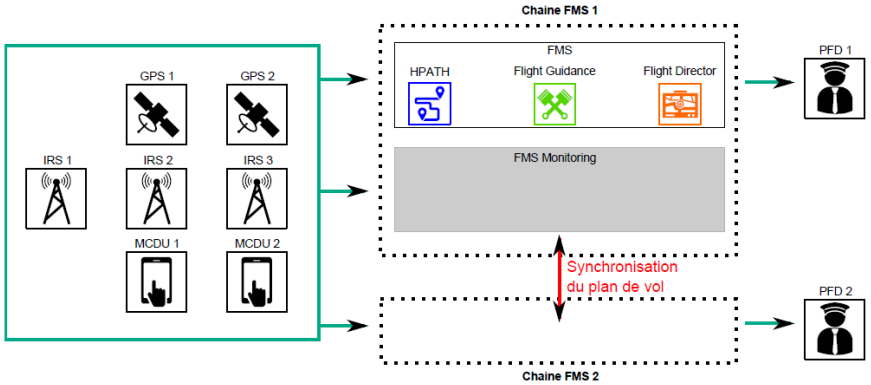


Figure 3: Principaux constituants de la chaîne de données de la fonction RNP-AR

Les standards aéronautiques (cf AMC 20-26, AC 90-101) fixent les exigences applicables à une fonction de navigation de qualité RNP-AR. Par exemple la précision des paramètres de navigation fournis doit être inférieure à 0,3 NM (nautical miles) en mode approche. La réglementation requiert aussi la disponibilité d'équipements précis (GPS, IRS, …) pour assurer un calcul d'approche RNP-AR.

On cherche sur ce cas d'étude à réaliser à l'aide de modèles GMD les étapes de conception amont suivantes :

- Evaluation de la sécurité d'une architecture de système de navigation de haute précision ;
- Calcul avant le décollage de l'avion des actions éventuelles de maintenance à effectuer pour pouvoir utiliser en toute sécurité le système de navigation (calcul de la minimum equipment list MEL) ;
- Spécification des tables de reconfiguration permettant d'adapter le comportement du système après l'apparition d'aléas.

4.2 Atelier de modélisation et d'analyse et application au cas d'étude

Un atelier de modélisation et d'analyse a été prototypé pour outiller le processus de conception à base de modèles GMD défini dans la figure 4.

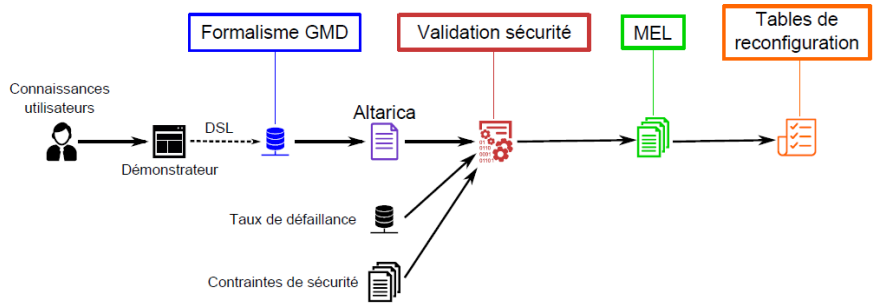


Figure 4: Processus outillé construit pour exploiter des modèles GMD

Un langage de domaine spécifique (DSL) et un éditeur associé ont tout d'abord été développés pour faciliter la capture des connaissances de l'utilisateur et produire des modèles GMD. Une interface graphique permet de visualiser la chaîne de dépendance entre ressources, leur mode et la propagation de la qualité de service comme dans les figures 1, 2 des exemples précédents. Les algorithmes de mise à jour de la qualité de service après changement de mode ont été aussi prototypés afin de simuler un modèle. Cet ensemble a permis de modéliser efficacement le cas d'étude et la figure 5 illustre la représentation graphique obtenue pour un sous-ensemble du cas d'étude.

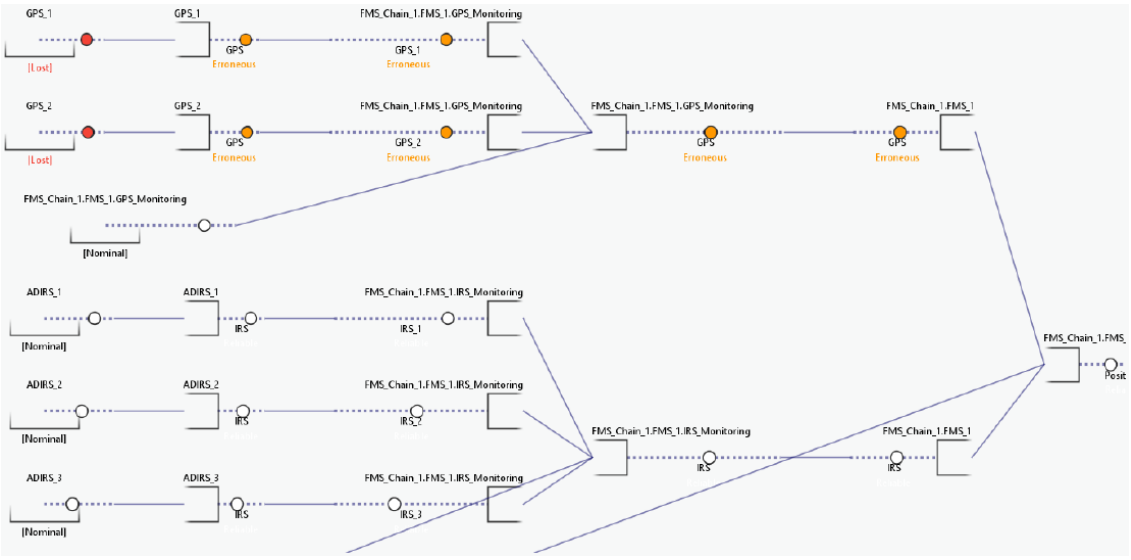


Figure 5: Vue graphique d'un sous ensemble du modèle GMD de la fonction de navigation

La traduction du modèle GMD vers AltaRica a été prototypée. Le modèle AltaRica produit peut alors être complété par les taux de défaillance des ressources et les cas de pannes à étudier pour évaluer le respect des contraintes de sécurité applicables à l'aide des outils existant par ailleurs.

Le calcul de la MEL peut être réalisé en utilisant de manière itérative les outils d'évaluation de sécurité pour déterminer si un système présentant une défaillance satisfait toujours les exigences de sécurité qui lui sont applicables et exiger des actions de maintenance si ce n'est pas le cas. Le calcul a été aussi étudié en recherchant des politiques d'actions de maintenance sur un processus décisionnel markovien généré à partir d'un modèle GMD du système qui inclut les actions de maintenance des ressources. Les politiques qui minimisent la probabilité d'atteindre un état où les qualités de service sont insuffisantes fournissent des résultats assistant le choix des réparations par les équipes de maintenance.

Le calcul des tables de reconfiguration a pour objectif d'indiquer au pilote les actions de reconfiguration interdites, possibles ou optimales après occurrence d'un aléa. Il est réalisable en recherchant des politiques d'action de contrôle des pilotes qui optimisent la qualité du service de navigation tout en respectant les contraintes de sécurité applicables. Ces politiques peuvent être calculée hors ligne et embarquées pour assister le pilote comme par exemple pour de nouveaux prototypes de système d'évitement de collision [13]. Cette fonctionnalité a pu être testée sur des exemples académiques mais pas sur le système de

navigation du cas d'étude. En effet, le système actuel contient beaucoup de modes de défaillances mais trop peu de modes de pilotage pour justifier l'emploi de calcul de politiques d'actions.

5. Conclusion

Cet article a présente le formalisme de gestion des modes dégradés GMD, la compilation de modèles GMD en modèles AltaRica et en processus décisionnel markovien ainsi qu'un atelier supportant l'édition et l'exploitation de modèles GMD. Les concepts et les outils ont été testés sur plusieurs cas d'étude académiques et sur un système de navigation pour l'atterrissage de haute précision plus représentatif de la complexité des systèmes aéronautiques actuels.

Ces expériences fournissent un premier retour positif sur la pertinence du formalisme GMD en phase amont de conception et sur la faisabilité d'outiller un processus de conception utilisant ce formalisme. En effet, GMD et les prototypes d'outils développés ou intégrés ont tout d'abord permis de modéliser de manière satisfaisante le système de navigation considéré. Les prototypes montrent aussi la faisabilité de la simulation de modèle GMD et de la recherche de combinaisons minimales de défaillance amenant à un événement redouté (après compilation en AltaRica). Ses fonctionnalités sont indispensables pour évaluer l'impact des défaillances en termes de capacité (qualité de service) et de sécurité. Nous avons prouvé que la génération automatique de tables d'aide à la décision était possible, en particulier concernant les conditions MEL et les décisions de reconfiguration. De plus les performances des outils de génération de table de décision ont été satisfaisantes dans des cas d'utilisation classiques.

Néanmoins, cette évaluation a aussi mis en avant des besoins d'amélioration de l'approche. Le formalisme GMD est actuellement trop restreint par rapport aux habitudes de modélisation des utilisateurs. Or les algorithmes applicables aux modèles AltaRica ou aux processus décisionnels markoviens peuvent traiter une classe plus large de modèles et ne justifient pas toutes les restrictions de GMD. Pour l'instant, le problème a été contourné en fournissant aux utilisateurs un langage de modélisation spécifique pour leur domaine (DSL) un peu plus riche que GMD. La réflexion doit être poursuivie pour soit étendre GMD, soit définir plus systématiquement les DSL pertinents.

La question du passage à l'échelle de la méthode proposée reste ouverte. Les exemples traités n'ont pas soulevé de difficulté en particulier parce que les décisions auxquelles nous nous intéressons (décision de conception, choix des reconfigurations...) ne s'intéressent qu'à des sous-ensembles du système complet. L'impact de la taille des modèles sur leur processus de capture et sur la performance des algorithmes devra être analysé sur les futurs systèmes mettant en œuvre des stratégies de reconfiguration dynamiques.

Enfin, les systèmes hautement dynamiquement reconfigurables n'existent pas aujourd'hui pour les fonctions les plus critiques d'un avion car leur complexité soulève des problèmes techniques difficiles. L'approche proposée fournit incontestablement des briques pour dépasser cette limitation et faciliter le développement de systèmes auto-adaptatifs critiques dans le futur.

Références

1. C. Papadopoulos, Ch. Seguin, P. Bieber, M. Bozzano, E. Boede, M. Bretschneider, A. Cavallo, R. Delmas, J. Deneux, J. Heckmann, O. Lisagor, M. Morel, L. Sagaspe, V. Sartor, Model-based safety assessment for the three stages of refinement of the system development process in ARP4754A , SAE 2011 AeroTech Congress & Exhibition, Toulouse, France, October 2011.
2. Papadopoulos Y., Walker M., Parker D., Rüde E., Hamann R., Uhlig A., Grätz U., Lien R. 2011. Engineering Failure Analysis & Design Optimisation with HiPHOPS, Journal of Engineering Failure Analysis, 18 (2): 590-608, Elsevier Science.
3. P. Bieber, R. Delmas, C. Seguin and M. Bretschneider 2011.b « Automatic derivation of qualitative and quantitative safety requirements for aircraft systems » ESREL 2011 September Troyes
4. P. Bieber , R. Delmas, C. Seguin. 2011.a DALculus: Theory and Tool for Development Assurance Level Allocation,Lecture Notes in Computer Science 6894:43-56, Springer.
5. Sorokos I., Papadopoulos Y., Azevedo L., Parker D., Walker M., 2015, Automating Allocation of Development Assurance Levels: an extension to HiP-HOPS, IFAC Dependable Control of Discrete Systems, Cancun, 2015.
6. J. Sprauel, A. Kolobov, and F. Teichteil-Königsbuch, 2014 Saturated path-constrained mdp : Planning under uncertainty and deterministic model-checking constraints. AAAI Conference on Artificial Intelligence, 2014.
7. J. Sprauel, C. Sannino, and C. Seguin, 2014, Techniques d'Aide à la Décision appliquées à la maintenance d'un avion de type Business Jet In Congrès Lambda Mu 19, DIJON, France, October 2014.
8. J. Sprauel, Conception sûre et optimale de systèmes critiques auto-adaptatifs soumis à des événements redoutés probabilistes, thèse de doctorat de l'université de Toulouse, Février 2016.
9. A. Arnold, G. Point, A. Griffault, and A. Rauzy. The altarica formalism for describing concurrent systems. Fundamenta. Informatica., 40(2-3) :109–124, November 1999.
10. T. Prosvirnova. AltaRica 3.0 : a Model-Based approach for Safety Analyses. Theses, Ecole Polytechnique, Novembre 2014
11. M. L. Puterman. Markov Decision Processes : Discrete Stochastic Dynamic Programming. John Wiley and Sons, Inc., New York, NY, USA, 1st edition, 1994
12. R.J. Kelly and J.M. Davis. Required navigation performance (rnp) for precision approach and landing with gnss application. Navigation, 41(1) :1–30, 1994.
13. Mykel J. Kochenderfer, Jessica E. Holland, and James P. Chryssanthacopoulos, Next-Generation Airborne Collision Avoidance System, LINCOLN LABORATORY JOURNAL, VOLUME 19, NUMBER 1, 2012

Office National d'Études et de Recherches Aérospatiales
2 avenue Edouard Belin - BP 74025
31055 TOULOUSE Cedex 4
Tél. : +33 5 62 25 25 25
http://www.onera.fr