

Rapprocher les études de sûreté de fonctionnement de l'ingénierie système : retour d'expérience

R. CRESSENT, V. IDASIAK, F. KRATZ

PRISME, UPRES 4229

ENSI de Bourges

Bourges, France

{robin.cressent ; vincent.idadiak ; frederic.kratz} @ensi-bourges.fr

Abstract— MéDISIS a été développé ces dernières années pour répondre à la problématique suivante : faciliter les études de sûreté de fonctionnement au sein de l'ingénierie système. Nous présenterons donc, dans cet article, la méthodologie d'analyse de la sûreté de fonctionnement des systèmes complexes, intégrée aux méthodes d'ingénierie système dirigées par les modèles nommée MéDISIS. Nous dégagerons l'apport d'un méta-modèle définissant les informations afférentes à la sûreté de fonctionnement, permettant à travers une base d'information : l'agrégation, la pérennisation et la traçabilité des connaissances des différents intervenants du projet. Ce modèle central est le socle du processus de traduction du modèle système vers les modèles supports des études de sûreté de fonctionnement ; ils constituent les différents processus MéDISIS. Nous appliquons MéDISIS dans le cadre du développement d'un système embarqué critique, que nous présenterons brièvement afin d'illustrer les gains obtenus lors des phases de spécification et de conception de ce dernier. Notamment, à travers ce projet industriel, nous insisterons sur les concepts propres à SysML permettant de préparer, piloter et manager les études de sûreté de fonctionnement.

Keywords : Ingénierie Système; sûreté de fonctionnement ; SysML ; AMDEC.

I. INTRODUCTION

L'ingénierie système est maintenant appuyée par des outils de plus en plus puissants, d'un point de vue langage de description, tel que SysML. Les plateformes professionnelles (Artisan Studio) ou libres (TopCased) fournissent des formats d'échange basé sur XML permettant l'extraction et la transformation des informations modélisées. Nous proposons, dans ce cadre technologique, un ensemble de processus de transformation de modèle facilitant la communication entre les activités d'ingénierie système pure et celle de la sûreté de fonctionnement. La méthode MéDISIS (Fig. 1), articulée autour d'une base de donnée, propose un ensemble de processus de traduction d'information, d'un modèle système central en SysML vers des langages de modélisation spécifiques afin de mener différentes études de sûreté de fonctionnement.

II. MEDISIS

Au cours du processus d'ISBM [4], le modèle commun d'ingénierie utilisé doit permettre d'initier et de connecter les

différentes activités d'analyse et de conception nécessaires à la production du système désiré.

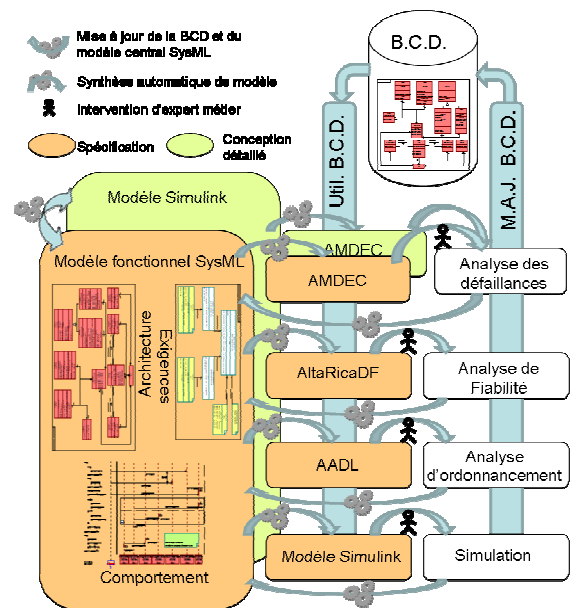


Figure 1. Schéma de l'ensemble des processus MéDISIS

Nous avons bâti une méthode afin d'intégrer efficacement le sous-processus d'analyse de SdF des systèmes au processus plus global d'ISBM. Cette méthode intitulée MéDISIS (Méthode D'Intégration des analyses de SdF à l'Ingénierie Système) a pour but de proposer un enchaînement de traitements de l'information et des connaissances, incluant leur création, expression, analyse, pérennisation et réutilisation répondant aux objectifs explicités ci-dessous. Ces traitements doivent être supportés par des outils et répertoires formant un EDS (Environnement de Développement Système) cohérent. Les objectifs à atteindre par la méthode sont les suivants :

1. Faciliter la transmission de connaissances entre équipes et entre les différentes activités d'ingénierie.
2. Accélérer la réalisation des études de SdF.
3. Organiser l'exploitation commune des connaissances sous forme de modèles.
4. Permettre la réutilisation des connaissances entre projets (i.e. favoriser le retour d'expérience).

5. Identifier les besoins d'analyse et réaliser le suivi de leurs résultats pendant les phases de vie du projet.
6. Améliorer la cohérence et la qualité des analyses SdF.

MéDISIS considère comme langage central le langage de modélisation système SysML [6] [7]. En effet, celui-ci, intrinsèquement profite aux points points 1 et 3 puisqu'il permet une modélisation multi-vue qui s'adaptera aux attentes des différents acteurs intervenants dans la conception du système. Un autre apport majeur de SysML est d'intégrer entre autres, la possibilité de modéliser les exigences en créant un support pour leur suivi au cours de l'évolution du modèle et du projet. Sur ce point, il satisfait donc l'objectif 5.

MéDISIS intègre, de manière centrale, aux différents processus une couche de persistance de l'information (BCD), celle-ci contribue à répondre aux attentes du point 1 en permettant une agrégation structurée et maîtrisée des connaissances en proposant à chaque expert, détenteur d'un point de vue, une structure centrale permettant la pérennisation des informations issues de ses analyses. La BCD apportera au point 3 son caractère multi-vue, et multi-langage et répondra aussi aux points 2 et 4 puisqu'elle est le résultat même de l'expression de ces besoins à savoir permettre un accès rapide aux informations de sûreté de fonctionnement issues du REX, aussi bien pour être exploitées que pour être archivées. Et enfin, la BCD permettra une cohérence des analyses de SdF grâce à l'architecture de son méta-modèle qui relie entre eux les informations fonctionnelles, métiers et les résultats des analyses ce qui répond en partie aux objectifs du point 6.

Enfin, MéDISIS se compose de plusieurs processus d'aide à la traduction permettant de passer d'un modèle en SysML vers d'autres modèles du même système dans des langages différents. Ce sont des processus automatisables, gages de cohérence, de rapidité et de traçabilité des informations traitées, créés et réutilisés, ils couvrent les besoins 1, 2 et 6. Ces processus de traduction doivent permettre de générer un squelette de modèle dans le langage cible le plus complet possible en respectant les points précédents. Ces processus ont pour but de permettre un déploiement opérationnel et de répondre aux besoins des industriels qui utilisent souvent plusieurs outils et formalismes au cours de leurs projets, manipulés par diverses personnes expertes dans leurs domaines propres.

À l'heure actuelle, MéDISIS propose plusieurs processus de traduction articulés autour d'un même langage source SysML. Les apports de SysML dans ce rôle sont décrits dans [2] et [3]. La mise en place de la BCD, le processus de génération d'AMDEC et la traduction de SysML vers Altarica sont décrits dans [3]. Dans [2], sont décrits les processus de traduction de SysML vers AADL et Simulink.

Nous allons dans la suite de cet article développer notre réflexion autour de l'apport d'un méta-modèle central permettant la généralisation de nos précédents travaux. Ce méta-modèle sera celui de la base de données qui devra permettre l'agrégation des connaissances d'entrée et de sortie des processus MéDISIS. Dans un premier temps, pour comprendre l'utilisation de ce méta-modèle dans notre

méthode, nous allons évoquer plus en détail le processus générique de traduction propre à MéDISIS.

III. LE PROCESSUS DE TRADUCTION

Ce processus de traduction mis en place par [3] pour la génération d'AMDEC à partir du modèle fonctionnel en SysML du système a été généralisé afin de décrire un processus de traduction générique entre un langage source et un langage cible.

Quels que soient les langages ciblés, certaines questions doivent être posées : existe-t-il des concepts en commun entre ces deux langages ? Existe-t-il des informations impossibles à retranscrire dans le langage cible ? Certaines informations impératives à la modélisation dans le langage cible sont-elles inexistantes dans le modèle du langage source ? Ces questions soulèvent le problème des domaines d'utilisation d'un langage de modélisation et de recouvrement des informations contenues dans un modèle. Deux paramètres sont donc importants à prendre en compte :

- Le périmètre de description du système (P) qui définira le type d'informations stockées par un langage : informations fonctionnelles, dysfonctionnelles, propriétés physiques, temporelles ou architecturales, comportements dynamiques...
- La précision de description du système (Pr) qui définira la qualité de l'information stockée par le langage dans un domaine (D) en particulier. Ce paramètre dépend de l'étape en cours dans le cycle de vie du système, la précision devant nécessairement augmenter avec l'avancement du projet (i.e. spécification, conception...).

Ces deux paramètres associés à un langage nous permettent de spécifier ce que l'on attend de ce processus de traduction.

Dans un processus d'ingénierie système, différents langages permettent de spécifier les systèmes à différents niveaux de conception et de faire le relais avec les outils d'analyse. Au fur et à mesure du processus de conception du système, les langages utilisés auront tendance à devenir de plus en plus spécialisés c'est-à-dire que leur périmètre va diminuer et leur précision augmenter. Un langage de modélisation système haut niveau tel que SysML possède un très large périmètre de description P, puisqu'il propose de nombreux concepts de modélisation tels que des *blocs* (représentant fonctions et composants, matériels ou logiciels), des interactions (solicitation de traitement, précedence d'activation...) des éléments architecturaux (connecteurs, ports, flux typés ou non, représentant les échanges de matières, énergie, données...), des diagrammes de type statechart pour l'aspect dynamique le plus formel. Il permet donc la modélisation de systèmes complexes. Il fournit un support important du suivi, de vérification et validation des exigences, basé sur le modèle hiérarchique des exigences et sur le mécanisme d'allocation. Cependant, beaucoup d'informations spécifiques, à des activités d'ingénierie, ne sont pas représentables en SysML ; si l'on se restreint à la norme, sans avoir recours aux mécanismes de stéréotypage et de définition de profils. Or, nous nous sommes fixés pour hypothèse

d'exploiter le langage SysML sans avoir recours à ces mécanismes afin de garder la possibilité d'interagir avec les outils sur étagère. Ainsi pour épauler l'ingénieur dans les processus d'IS, les ponts entre langages que nous avons mis en place permettent de générer des modèles de périmètres moins vastes mais avec une plus grande précision de description afin de suivre, le processus de vérification et de conception d'un système. On peut, notamment, citer l'AMDEC qui décrit les aspects dysfonctionnels d'un système, système mais ne considère que peu d'autre aspect du système, ou encore AADL qui, quant à lui, reste un outil de niveau assez élevé mais dispose cependant d'une précision importante pour ce qui est du domaine des contraintes et propriétés temporelles, ainsi que l'allocation des fonctionnalités hardware et software d'un système de contrôle.

Lors d'un processus de traduction, l'ensemble des informations est disponible dans le modèle source et est accessible selon les points d'entrée et le méta-modèle sous-jacent. Par contre, il est évidemment rarement possible de renseigner, par copie ou déduction, tous les éléments du langage cible. En effet, le modèle cible est là pour étendre ou raffiner les possibilités du modèle actuel du système. De nouvelles informations sont donc nécessaires que ce soit des informations d'un domaine différent ($P(\text{Cible}) \not\subset P(\text{Source})$) ou d'un niveau de détail plus grand dans un domaine particulier ($\text{Pr}(\text{Cible})_D > \text{Pr}(\text{Source})_D$, où D est un domaine quelconque). Lorsque des informations sont inexistantes dans le modèle source, il est nécessaire de :

- Déduire ces informations de celles contenues dans le modèle source.
- Faire appel à une source extérieure, un expert du domaine, ou avoir recours à une base de données.

Dans le cas où des données nouvelles non déductibles du modèle source ont été apportées par une source extérieure, comme nous l'avons spécifié dans les objectifs de la méthode MéDISIS, il est utile de mettre en place un moyen de pérenniser ces informations afin de les réutiliser lors de projets suivants. C'est pourquoi une base de données implantée au centre du processus de traduction a été définie. Cette base de données permet de mettre à disposition des informations métier nécessaires au processus de traduction, afin de limiter la nécessité d'avoir recours à un expert. Et si un expert est nécessaire malgré tout, elle permet de stocker les informations apportées par celui-ci afin d'être plus efficiente lors du projet suivant.

Enfin, pour mettre au point notre processus de traduction d'un langage source vers un langage cible avons identifié 4 étapes successives :

1. Rédiger la table d'équivalence. Cette table permettra de relier entre eux les différents concepts propres à chaque langage.
2. Identifier, les informations déductibles du modèle source, et la façon de les déduire.
3. Dimensionner, la base de données, pour les informations nécessaires au langage cible et qui ne sont pas déductibles, le méta-modèle garantissant alors une

croissance raisonnée et cohérente.

4. Identifier et guider l'introduction des données de l'expert pour compléter les ressources fournies par la base de données et ainsi en permettre sa mise à jour.

L'étape 1 est réalisée en effectuant une analyse complète du méta-modèle des langages source et cible. Cette étape peut aussi être l'occasion de définir le méta-modèle du langage s'il n'est pas disponible ou dépend de l'étude (cf. Méta-Modèle de l'AMDEC, [3]). L'étude du méta-modèle permet de lister l'ensemble des concepts représentés dans un langage, d'identifier qualitativement l'information et permet de définir les moyens d'y accéder. Ainsi l'analyse du méta-modèle des langages source et cible, si elle est suivie d'une étude comparative, permet de rédiger le tableau des équivalences. De plus, la comparaison des concepts de chaque langage met en avant les manques d'équivalence qui peuvent exister entre eux deux. Ces manques et les actions nécessaires à leur circonscription constituent les objectifs de la deuxième étape.

L'étape 2 doit permettre de relier certains concepts présents uniquement dans le langage cible à des informations modélisées dans le langage source. Ces relations de déduction seront propres à chaque couple langage cible/langage source. Une fois la logique de déduction établie, on pourra représenter cette relation sous forme de « parametric diagram », ainsi l'algorithme pourra être mis en commun au sein de la BCD. Cependant, il faudra donc prévoir au sein de notre BCD les moyens d'intégrer de façon cohérente ces relations, ceci est le but de la troisième étape.

L'étape 3 est primordiale pour la pérennisation des informations générées par un processus MéDISIS et pour inscrire cette méthode dans un cycle d'itérations successives d'amélioration de l'outil. En effet, de la bonne structuration de la BCD dépend l'efficacité de l'agrégation des connaissances métiers et de l'évolution de chaque processus de traduction qui compose MéDISIS. Car, c'est en modélisant efficacement les informations métiers au sein de la BCD que les experts seront dégagés des traitements de reprise de modèle fastidieux et source d'erreur pour se concentrer principalement sur leurs études. Pour remplir ces objectifs, le méta-modèle de la BCD doit être :

- Complet, et ainsi former un noyau commun d'information métier, organisé et accessible reprenant les concepts issus de l'analyse des méta-modèle des langages cible.
- Structuré, de façon à modéliser toutes les relations et échanges possibles entre tous ces concepts afin de pourvoir à tous les besoins des différents processus de traduction mis en oeuvre.
- Accessible, pour permettre son extension naturellement à la modélisation d'un nouveau type de données (nécessaire à un langage cible), et pour permettre à la BCD d'organiser les informations d'un grand nombre d'entités.

Finalement, l'étape 4 souligne les informations manquantes du modèle cible, orientant ainsi le travail de l'expert. Pour

certain processus (AMDEC et Altarica), la mise à jour du contenu de la BCD est également réalisée lors de cette étape à partir des informations injectées par l'expert.

Les « parametric diagrams » de SysML possèdent la qualité nécessaire à l'expression de ces analyses [1] [2] [3]. La Fig. 2 illustre notamment l'analyse temporelle du système LEA ; il pérennise et introduit au niveau système, les résultats génériques de l'expertise du système embarqué temps réel.

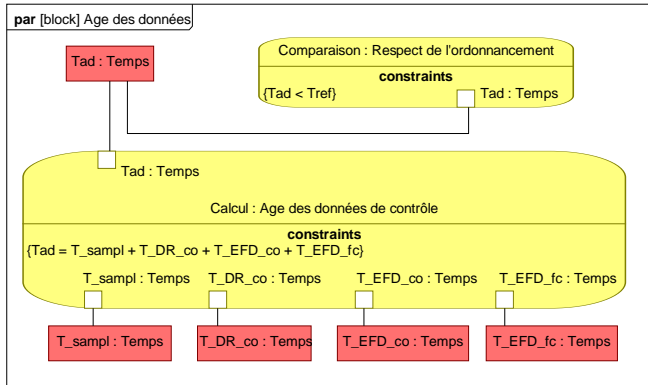


Figure 2. « Parametric diagram » décrivant la formule d'évaluation de l'âge des données.

IV. UTILISATION ET RETOUR D'EXPERIENCE

MéDISIS est actuellement utilisée dans le cadre d'un partenariat avec la société MBDA, afin de réaliser le système de contrôle de vol d'un véhicule hypersonique. Notre équipe a en charge le déploiement d'une méthodologie d'ingénierie système facilitant les études de sûreté de fonctionnement et le développement du système embarqué contrôlant la séquence de vol automatique du système LEA. C'est pourquoi nous utilisons MéDISIS et avons initié de nouveaux processus (i.e. AADL, Simulink). Le système LEA [5] doit servir à tester un nouveau type de propulsion en conditions réelles. Pour ce faire, notre partenaire a décidé la conception d'un véhicule hypersonique, largable depuis un avion qui permettra l'observation par télémétrie du comportement du réacteur. De par sa nature technologique et ses besoins fonctionnels, le projet LEA offre, de nombreux cas d'étude en fiabilité et en sûreté fonctionnelle. En effet, nous devons garantir certaines fonctionnalités telles que : les autotests, la fonction de détection du largage, les consignes moteur, la détection de fin de mission et la télémétrie. Dès les premières étapes, du projet, nous avons constaté l'intérêt de rapprocher l'IS des études de sûreté de fonctionnement.

A. Le processus d'ingénierie système

Plus particulièrement, nous relevons, ici, les avantages rencontrés lors de la phase d'analyse et de conception préliminaire du projet. Le processus suivi est le suivant :

-1- *Elicitation des connaissances projets à partir des spécifications techniques du projet.* Cette étape suit un processus d'ingénierie classique comprenant : la formalisation des exigences avec les diagrammes d'exigences de SysML, la classification des besoins grâce au diagramme de cas d'utilisation, la synthèse des spécifications techniques du

partenaire, par cas d'utilisation, avec des diagrammes de séquence. La constitution des diagrammes paramétriques qualifiant les contraintes environnementales et techniques permet la modélisation des paramètres physiques dimensionnant du système. Cette étape se termine par la définition de la vue organique du système et l'allocation des exigences aux différentes vues du modèle (Fig. 3).

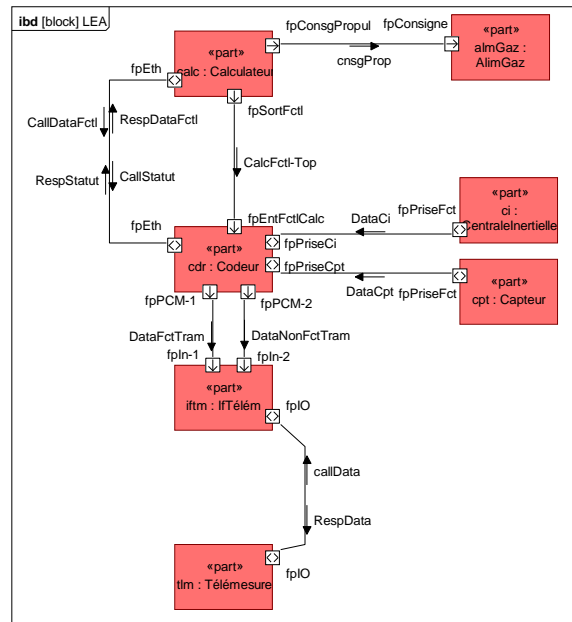


Figure 3. Internal block diagram du véhicule LEA

-2- *Analyse système.* L'étape d'analyse système est réalisée, l'apport principal de SysML étant de réifier les analyses par le biais de « parametric diagram » et l'établissement de liens (i.e. binding SysML) entre les éléments du modèle à travers ses différents points de vue. On peut parler de synchronisation et de mise en cohérence de l'information.

-3- *Analyse de risque.* L'étape analyse de risques débute par la synthèse d'une pré-AMDEC, grâce au processus AMDEC de MéDISIS. Le projet étant nouveau, les modes de défaillances génériques sont, dans un premier temps appliqués. Au fur est à mesure de l'avancement de l'analyse, les résultats sont introduits dans la BCD pour une définition plus judicieuse des modes de défaillance des composants. Ce qui permet alors à chaque modification du modèle système, la synthèse rapide d'une nouvelle pré-AMDEC [3]. Une liste de points sensibles est dégagée, identifiant les risques les plus élevés avec les composants et fonctions impactés, ainsi que les exigences pouvant être atteintes. Le nombre et la nature des exigences impactées contribuent évidemment, à la cotation de la criticité du mode de défaillance

-4- *Analyse système.* L'intégration des résultats de l'analyse de risque, conduit à renforcer (modifier) les exigences (diagramme d'exigences) et les contraintes (« parametric diagram ») du système (modèle). À l'issue de cette nouvelle synthèse, plusieurs choix d'architectures se présentent. Les critères de discrimination sont de nature fonctionnelle, économique, sécuritaire et temporelle. Certains choix architecturaux combinant les aspects fonctionnels et de sûreté

Nom	Modes de défaillances	Causes	Effets locaux	Effets exigences	Effets systèmes
Calculateur	Défaillance d'ordonnement	Flux Ethernet [Contrainte Env : vibration]>[Specif. Connecteur]	Consigne de propulsion [AlimGaz] / Sorties Fonctionnelles [Codeur]	Contraintes temps réelles non respectées	Perte de données capteurs non transmises au codeur destinées à la télémesure / Risque de mauvais fonctionnement du moteur si les consignes ne sont pas émises convenablement.
		Surcharge Interne	Consigne de propulsion [AlimGaz] / Sorties Fonctionnelles [Codeur]	Contraintes temps réelles non respectées	Perte de données

Figure 4. Extrait de l'AMDEC du système LEA

fonctionnelle sont classiquement résolus à ce stade tel que le placement de la centrale inertielle du véhicule [2].

-5- *Analyse spécifique métier.* Afin de résoudre les points relevés par l'AMDEC associant des critères plus spécifiques tels que les traitements temps réel, il est nécessaire de déclencher le processus MéDISIS ad hoc. Ici, le processus AADL décrit dans [2].

À ce stade, le processus est répliqué de l'étape 1 à 4 mais pour un domaine particulier celui des architectures temps réel. L'intérêt des modèles AADL, pour ce type d'étude, a été discuté précédemment, notons simplement que les résultats : modèles des contraintes temporelles, choix des architectures matérielles et logicielles de traitement sont également retranscrits dans les modèles système.

-6- *Injection des modèles de défaillance.*

Pour permettre la passerelle vers le domaine de la conception détaillée, nous utilisons les similitudes relevées par [9] entre SysML et Simulink afin d'obtenir la base du modèle de conception. C'est également à ce stade que nous injectons dans le modèle du système, les modèles des modes de défaillances les plus critiques afin de réaliser les simulations du comportement du système et valider les choix de conception vis-à-vis des exigences de sûreté.

B. *Retours d'expérience*

Dans ce paragraphe, nous illustrons ces résultats à travers l'étude de la connexion entre le calculateur de vol du véhicule et le module d'acquisition des données.

Les diagrammes (Fig. 3 et Fig. 5) appartiennent au modèle source d'un système modélisé en SysML, nous avons représenté ici l'« Internal Block Diagram » de notre système ainsi que le Diagramme de Séquence spécifiant la transmission Ethernet/IRIG que l'on souhaite étudier en détail. L'IBD figure 2 exprime l'architecture du système c'est-à-dire la vision organique choisie à ce stade du projet afin de permettre le contrôle commande du véhicule et la télétransmission des 300 capteurs. Il faut noter l'intérêt des IBD dans ce cas, où les flux entre les composants du système sont typés en fonction de leur qualification électrique, nature et quantité de donnée ou énergie les traversant. Ces types sont supportés par des « parametric diagram » lorsqu'ils sont paramétrés par des grandeurs temporelles ou des grandeurs physiques. Ce sont ces grandeurs que doit optimiser l'ingénieur système. Ce sont également ces grandeurs qui sont inspectées par les études de sûreté de fonctionnement.

À travers l'analyse fonctionnelle ramenée ici à la vue structurelle de notre système (Fig. 3) (les fonctions sont liées aux composants au travers des diagrammes d'activités ou par le biais des diagrammes de séquences lorsque le temps est prédominant) ; nous pouvons définir les paramètres d'influence de la connexion Calculateur – Module d'acquisition. Ces paramètres, exprimés sur le diagramme de séquence (Fig. 5) correspondent aux temps de traitements unitaires¹.

Grâce à la génération d'AMDEC nous pouvons expliciter un mode de défaillance lié à nos exigences temps réels (Fig. 4). On constate que le flux Ethernet/IRIG entre le module d'acquisition et le calculateur peut être la cause d'une défaillance grave du système issue du non-respect des contraintes temps réels. Ces résultats nous orientent vers une étude plus poussée du respect de ces contraintes liées à ce type d'architecture de contrôle et à ce type de technologie. Nous allons donc ensuite traduire notre modèle fonctionnel vers un modèle AADL qui est un langage prévu pour la modélisation de systèmes hybrides notamment les systèmes avionique embarqués.

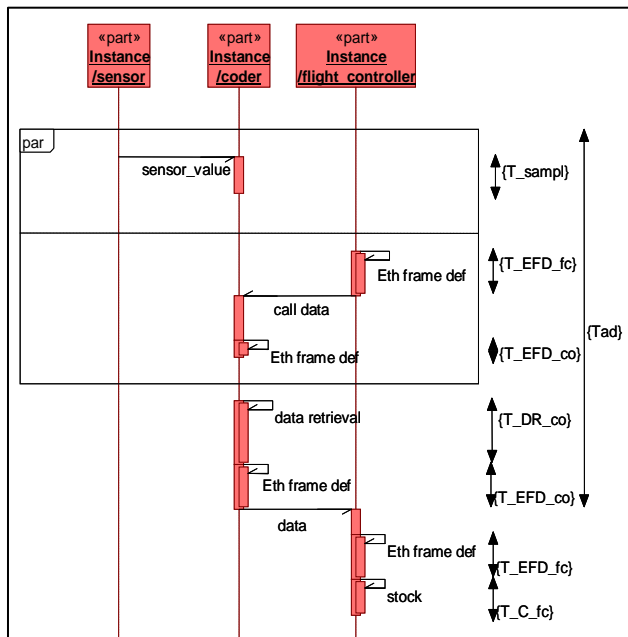


Figure 5. Diagramme de séquence de la connexion Ethernet entre le calculateur et le module d'acquisition

¹ À chaque étape de l'ingénierie système, un niveau de détail est fixé, suivant la complexité et la nature du projet, notamment l'emploi de COTS ou non. Dans ce projet, l'emploi de COTS logiciel et matériel est important, il fixe naturellement un niveau de détail pour la conception préliminaire. Bien qu'hétérogène, ce niveau de détail permet l'établissement du temps de réalisation des contrôles et temps de traitement des données que nous appelons « temps unitaires ». Ces temps de traitement sont dans une première étape extraits des spécifications techniques et sont les premières grandeurs réévaluées lorsque la maquette fonctionnelle du système est réalisée.

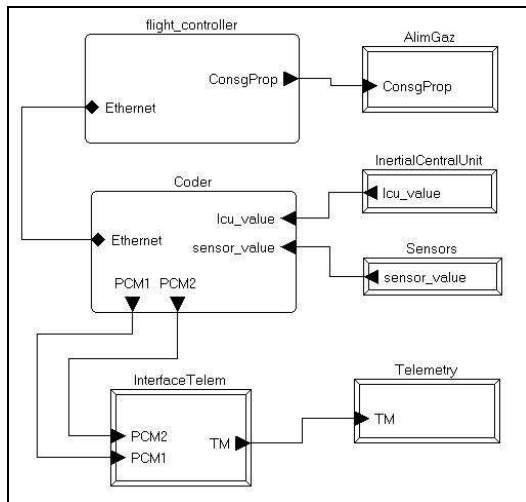


Figure 6. Modèle AADL du système LEA

Le modèle AADL du système obtenu (Fig. 6) présente une hiérarchie proche du modèle SysML source, cependant chaque composant est ramifié en sous-composants matériels et logiciel. Nous avons du introduire pour la traduction les informations métiers (i.e. applications temps réels) essentielles à l'établissement d'une architecture embarquée. À partir de la description des processus gérés par notre ordinateur et le module d'acquisition, nous pouvons faire appel à Cheddar [8], notamment pour analyser l'ordonnancement des tâches au sein des processeurs du ordinateur et du module d'acquisition. Cette analyse temporelle permet d'établir la formule d'évaluation de l'âge des données et le temps limite à ne pas dépasser. L'âge des données correspond à la durée temporelle mesurable entre la production d'une grandeur physique issue de la lecture d'un capteur et sa mise à disposition pour un algorithme de contrôle dans le ordinateur, en traversant les couches protocolaires Ethernet et IRIG de la connexion Ordinateur – Module d'acquisition. La formule d'évaluation est stockée dans la BCD à l'aide de « parametric diagrams » (Fig. 2).

Enfin, au moment de passer à la conception détaillée en générant une architecture à compléter en Simulink, on peut au final obtenir les évolutions simulées de nos temps d'influence et vérifier le respect des contraintes au cours de la simulation. Dans ce modèle Simulink, nous intégrons les défaillances listées dans l'AMDEC afin de simuler le comportement de notre système dans ces cas particuliers. L'injection de fautes nous permet de décliner les différents niveaux d'acceptation du risque et de ségrégation du risque. Notamment, par l'injection de défaillances, nous avons pu établir l'utilité d'une séquence standard de déroulement des consignes gaz qui permet d'assurer un contrôle du véhicule minimum en cas de perte partielle des informations capteurs.

Avant implémentation du code sur le processeur, l'ingénieur système peut considérer que l'équation temporelle est fixée par le « parametric diagram » aux conditions suivantes : la dernière valeur d'un capteur a été mémorisée, une

séquence de contrôle standard est implémentée, la retransmission des trames de télémesures n'est demandée qu'à la deuxième perte consécutive d'une trame.

Relevons alors, que ce « parametric diagram » constitue un des premiers tests d'acceptation de la cible temps réel (i.e. processeur et noyau). De même, les jeux de tests de la robustesse de notre système reprendront le modèle de faute testé dans Simulink pour achever la validation du système embarqué. En effet, les paramètres du test tels que les signaux et/ou trames de transmission, ainsi que leur forme perturbée pourront être générés à partir du modèle de faute et injectés sur la cible. Un dispositif de type générateur de signaux permettra la génération de ces entrées, et la génération des trames. Ce procédé permettra de ne reprendre que les jeux de tests les plus significatifs relevés au niveau modèle.

V. CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons décrit le processus d'ingénierie système permettant de tirer profit de MéDISIS au sein d'un projet industriel. L'accent a été mis sur le rôle central de la BCD en illustrant notamment comment SysML permet de thésauriser le retour d'analyse des processus métier à travers les « parametric diagrams » et les IBD. Dans ce projet, MéDISIS s'intègre efficacement dans une stratégie d'ingénierie dirigée par les modèles, nous sommes actuellement en train de quantifier les gains en temps de conception. Nos travaux s'orientent également vers la particularisation de MéDISIS pour les systèmes à COTS, afin d'intégrer la gestion des différents niveaux de description et de validation des composants.

RÉFÉRENCES

- [1] B. Cole, C. Delp & K. Donahue, "Piloting model based engineering techniques for spacecraft concepts in early formulation", California Institute of Technology, published by INCOSE, 2010.
- [2] R. Cressent, P. David, V. Idasiak & F. Kratz, "Increasing Reliability of Embedded Systems in a SysML Centered MBSE Process: Application to the LEA Project", 1st M-BED workshop, during DATE 2010, Dresden, Germany, 12 March 2010.
- [3] P. David, V. Idasiak, et F. Kratz, "Reliability study of complex physical systems using SysML", Journal of Reliability Engineering and System Safety, Volume 95, Issue 4, April 2010, Pages 431-450.
- [4] J. Estefan. "Survey of Model-Based Systems Engineering (MBSE) Methodologies", Rev. B. INCOSE MBSE Initiative, 23 Mai 2008.
- [5] F. Falempin, L. Serre, "French Flight Testing Program LEA Status in 2009", 16th AIAA/DLR/DGLR International Space Planes and Hypersonic Systems and Technologies Conference, Bremen, Germany, Oct. 19-22, 2009
- [6] S. Friedenthal, A. Moore & R. Steiner, "A Practical Guide to SysML : The Systems Modeling Language", The MK/OMG press, Elsevier, 2008.
- [7] OMG 2008. "OMG Systems Modeling Language" (OMG SysML) V1.1., 1st November 2008.
- [8] F. Singhoff, "The Cheddar AADL Property sets (Release 2.x). LISyC technical report, February 2007
- [9] R. Snyder, D. Bocktaels, X. Feigentaels, "Validation fonctionnelle à l'aide d'une transformation SysML/Simulink", Journées de travail Neptune No7, Toulouse, FRANCE, 18/05/2010, pp. 49-53.