



**HAL**  
open science

# Prise en compte des analyses de la sûreté de fonctionnement dans l'ingénierie de système dirigée par les modèles SysML

Robin Cressent, Vincent Idasiak, Frédéric Kratz

## ► To cite this version:

Robin Cressent, Vincent Idasiak, Frédéric Kratz. Prise en compte des analyses de la sûreté de fonctionnement dans l'ingénierie de système dirigée par les modèles SysML. Génie logiciel: le magazine de l'ingénierie du logiciel et des systèmes, 2011, p33-39. hal-00630950

**HAL Id: hal-00630950**

**<https://hal.science/hal-00630950>**

Submitted on 11 Oct 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Prise en compte des analyses de la sûreté de fonctionnement dans l'ingénierie de système dirigée par les modèles SysML

Robin CRESSANT, Vincent IDASIAK et Frédéric KRATZ

**Résumé :** La méthode MéDISIS a été développée, afin de faciliter les études de sûreté de fonctionnement, au sein de l'ingénierie de système dirigée par les modèles. Cet article indique comment, à travers une modélisation des processus d'ingénierie de système et de sûreté de fonctionnement, sont déployés MéDISIS et ses processus. Il souligne également l'apport du méta-modèle définissant les informations afférentes à la sûreté de fonctionnement, permettant à travers une base d'informations l'agrégation, la pérennisation et la traçabilité des connaissances des différents intervenants du projet. MéDISIS appliquée au développement d'un système embarqué critique illustre ainsi les gains obtenus lors des phases de spécification et conception de ce dernier. À travers ce projet industriel, sont mis en évidence les concepts propres à SysML permettant de préparer, piloter et conduire les études de sûreté de fonctionnement.

**Mots clés :** Ingénierie de système, sûreté de fonctionnement, SysML, AMDEC.

## 1. INTRODUCTION

La réalisation de systèmes innovants modernes nécessite l'emploi de processus à même de piloter le projet, de l'expression des besoins au déploiement du système ; ils sont désignés par le terme de processus d'ingénierie de systèmes (IS) [4,6]. Les processus organisent l'avancée à travers les niveaux de détails et de composition, tout en ayant pour objectifs l'assurance de la pérennisation des exigences, décisions et connaissances. La dimension sûreté de fonctionnement (SdF) doit être intégrée plus intimement à l'IS mais est encore trop indépendante [3]. La méthode MéDISIS a pour objectif de favoriser l'interaction nécessaire entre l'IS et la SdF.

Lors de précédents travaux, les processus MéDISIS ont été définis en intégrant les études de SdF dans l'IS de partenaires industriels [2]. La première partie de cet article présente (§2) comment et à quel niveau de granularité sont modélisés les activités d'IS et les besoins en SdF, des parties prenantes. De ce point, sont dégagés les besoins que les processus MéDISIS doivent couvrir en termes de synchronisation des données et thésaurisation des connaissances. Ils seront repris au § 3 où est introduit également le méta-modèle central de MéDISIS permettant de définir la Base de données des Comportements

Dysfonctionnels (BCD). Dans le §4, les processus actuels de MéDISIS sont présentés à travers la définition de l'environnement de développement système (EDS) utilisé. Finalement, l'apport de MéDISIS est illustré (§5), à travers le retour d'expérience d'un projet industriel, pour la prise en compte de la sûreté de fonctionnement dans l'IS.

## 2. MODELES DES PROCESSUS IS ET SdF

Nous avons choisi de modéliser uniquement les activités d'IS et de SdF, et de fournir un support générique transposable aux différents processus IS et de SdF complets. L'appropriation par les partenaires industriels en est facilitée, chacun devant instancier ceux-ci dans son propre référentiel. Ce niveau de détail est cependant suffisant si l'on qualifie la nature et le niveau sémantique des informations résultantes des activités, ce que réalisent les référentiels d'entreprise. Le modèle UML (Figure 1) reprend donc les activités d'IS et de SdF, sous la forme d'activités, et reprend les informations générées par ces activités sous la forme d'entrepôt de données. Ici, les entrepôts de données modélisent une vue ou une partie des informations disponibles dans un modèle donné. Ils correspondent à la définition que le standard UML 2 [8] leur donne : « Les données contenues dans un entrepôt de données sont persistantes et utilisées en fonction du besoin des activités ».

La figure 1 représente la modélisation d'une partie des activités d'IS, celles qui correspondent aux premières phases du développement d'un système :

- Définir les objectifs du système.
- Établir ses fonctionnalités.
- Établir les performances requises (exigences et contraintes).
- Développer l'architecture physique et logique.

Ces quatre activités vont générer des informations qui seront aisément modélisées en ayant recours à SysML. Ces informations peuvent être représentées par un entrepôt de données. Les informations de chaque entrepôt de données sont issues de certains diagrammes SysML (Figure 1) et qualifient les ensembles d'entrée des fonctions de transfert que devront assumer les processus MéDISIS.

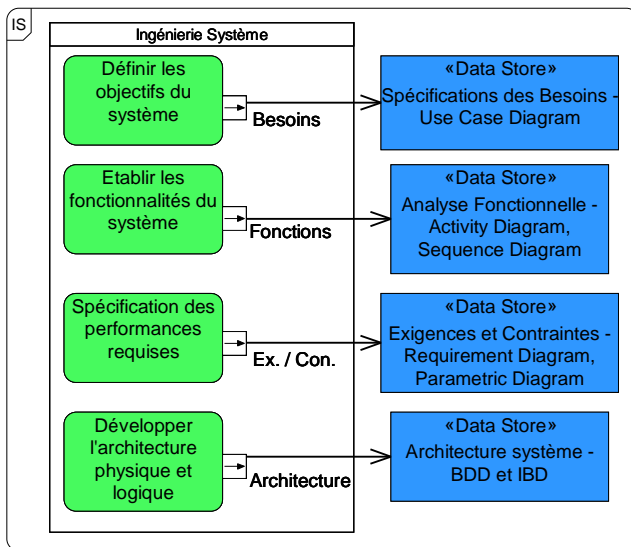


Figure 1 : Processus d'ingénierie de système

L'absence de flux de contrôle est due à la granularité de modélisation choisie correspondant au choix de mettre au point un modèle suffisamment générique pour qu'il reste applicable dans la plupart des référentiels industriels. Ces activités peuvent être réalisées en parallèle ou en cascade selon le contexte industriel. Il en va de même pour les itérations nécessaires de ces activités, les données des entrepôts de données sont simplement écrasées, la gestion de version étant allouée à la BCD définie dans le paragraphe 3.

Afin de définir ce que doivent produire les processus MéDISIS, les activités de la SdF sont également modélisées. Ici, seul le processus d'analyse, menant à la rédaction d'une AMDEC, est modélisé (Figure 2). Dans ce diagramme d'activité, le seul flux de contrôle présent correspond au choix initial de l'expert d'effectuer une AMDEC fonctionnelle ou une AMDEC composants. Les entrepôts de données du diagramme d'activité (Figure 2) modélisent les besoins (informations nécessaires) pour réaliser une AMDEC. Les entrepôts de données « Analyse Fonctionnelle » et « Dossier de Conception » sont naturellement issus des entrepôts de données du processus d'IS. L'entrepôt de données « REX » peut être en partie couvert par la BCD de MéDISIS.

« L'arborescence des fonctions du système » nécessaire au processus d'AMDEC pourrait être satisfaite par « l'analyse fonctionnelle » modélisée en diagramme d'activité et diagramme de séquence qui résultent des activités d'IS, de même pour l'architecture hiérarchisée des composants du système et l'architecture système modélisée en diagramme de définition de blocs et diagramme interne de bloc. De plus, les exigences et contraintes du système sont utiles au processus d'AMDEC

pour aider l'expert à évaluer la criticité d'un mode de défaillance. Ces constats définissent le cahier des charges des processus d'automatisation du transfert d'information entre IS et SdF.

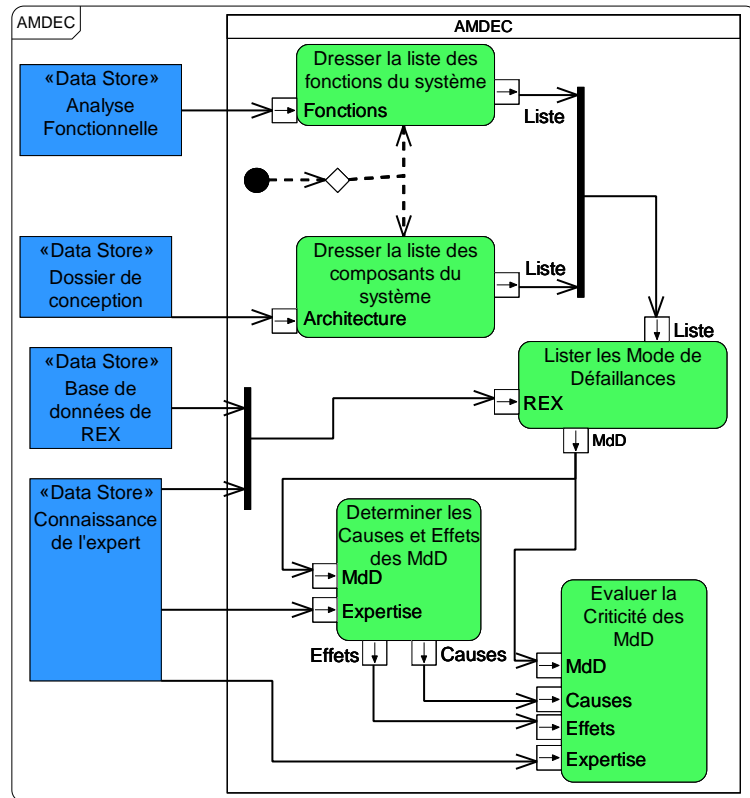


Figure 2 : Processus d'AMDEC

Dans le cas particulier du processus de rédaction d'AMDEC, nous avons été plus loin qu'une simple traduction, puisque nous avons mis au point un processus automatisé de génération d'un rapport d'AMDEC préliminaire. Ce que l'on définit par « rapport d'AMDEC préliminaire » est un tableau généré à partir de toutes les informations disponibles à la suite des activités d'IS ; ce tableau doit permettre à l'expert de gagner du temps afin de concentrer son intervention sur la complétion de cette AMDEC préliminaire. Celle-ci met à disposition de l'expert la liste de tous les composants du système, la liste de toutes les fonctions du système, les modes de défaillances associés (présents dans une base de données de REX) ainsi que toutes les causes envisageables compte tenu de l'architecture du système, ainsi que les effets potentiels au niveau local, système ainsi que sur les exigences. Afin que ce processus soit constamment amélioré par chaque projet réalisé, il est prévu de mettre à jour la base de données afin qu'elle contienne toujours plus de modes de défaillance répertoriés. Ainsi, l'apport de MéDISIS pour la génération d'AMDEC peut être modélisé selon la figure 3 et est détaillé dans [3].

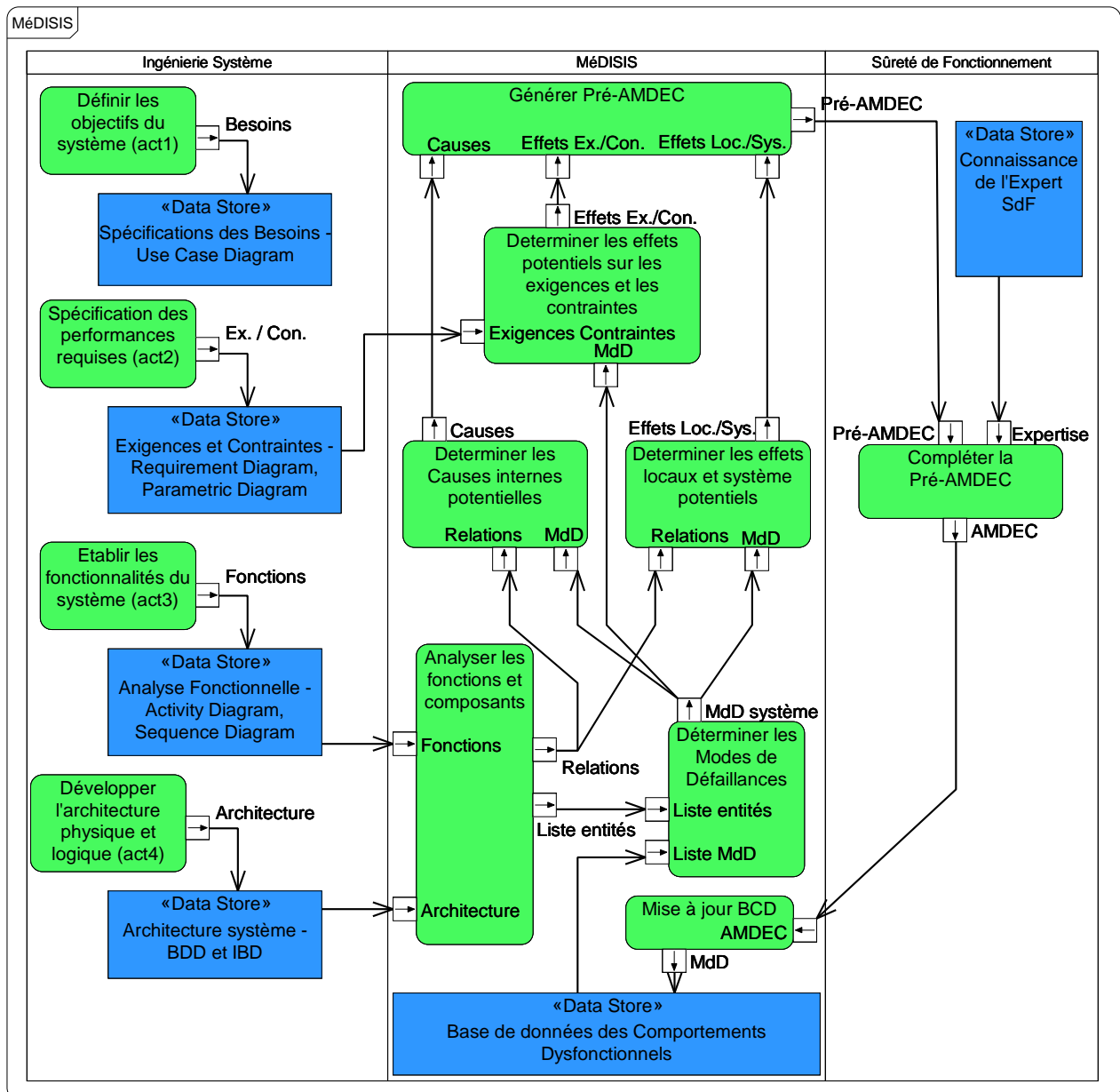


Figure 3 : Activités de MéDISIS pour la génération d'AMDEC

Ce premier processus de rédaction d'AMDEC préliminaire permet de dimensionner les besoins en matière de thésaurisation des informations dysfonctionnelles qui doivent être satisfaits par la BCD.

### 3. LA BASE DE DONNEES DES COMPORTEMENTS DYSFONCTIONNELS

Cette base (BCD) doit remplir deux objectifs :

- Thésauriser les informations de Retour d'Expérience (REX).
- Assister les processus MéDISIS.

La BCD doit permettre de stocker de façon organisée les modes de défaillances, ainsi que le comportement du composant impacté. Pour cela, un premier méta-modèle de la BCD en SysML a été établi (Figure 4).

Les modes de défaillances héritent du type de composant, de sorte qu'ils constituent une redéfinition des composants intégrant une partie dysfonctionnelle. Les propriétés du composant sont reprises ; ainsi elles pourront être redéfinies par rapport à une vue dysfonctionnelle ou réutilisées lors de la définition du comportement défaillant du composant. La définition du comportement dysfonctionnel est nécessaire pour déduire les effets locaux des modes de défaillances déclarés. Nous proposons l'utilisation de deux types d'éléments venant s'associer : les *opérations* et les *machines à états finis*. Les

opérations sont utilisées comme actions d'état dans la machine à états finis. Cette dernière permet de décrire le ou les changements d'état menant à la défaillance du composant. Les actions d'état définies en entrée (entry : action) transcrivent l'effet de la défaillance au moment du changement d'état. Les actions réalisées dans l'état défaillant (do : action) modélisent le traitement accompli par le composant dans cet état de défaillance.

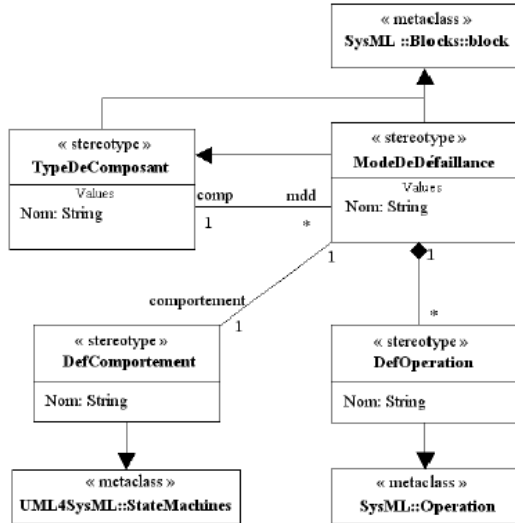


Figure 4 : Métamodèle de la BCD pour la génération d'AMDEC préliminaire

L'ajout de processus à MéDISIS a été réalisé selon la méthode précédente : modélisation IS, modélisation activité SdF, incrément méta-modèle BCD, ce qui permet de modéliser efficacement les informations métiers au sein de la BCD. Les experts seront ainsi dégagés des traitements de reprise de modèle fastidieux et sources d'erreur pour se concentrer principalement sur leurs études. Le méta-modèle de la BCD est alors :

- Complet, et ainsi forme un noyau commun d'informations métier, reprenant les concepts issus de l'analyse des langages propres à la SdF.
- 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 œuvre.
- Accessible, pour permettre naturellement son extension à la modélisation d'un nouveau type de données et pour permettre à la BCD d'organiser les informations d'un grand nombre d'entités.

#### 4. L'ENVIRONNEMENT DE DEVELOPPEMENT SYSTEME (EDS)

Finalement, à partir de ces premières réflexions concernant le processus de rédaction d'AMDEC, a été décrit un ensemble de processus de transformation de modèles facilitant la communication entre les activités d'ingénierie de système pure

et celle de la sûreté de fonctionnement. La méthode globale MéDISIS (Figure 5), articulée autour de la BCD, propose un ensemble de processus de traduction d'informations, d'un modèle système central en SysML vers des langages de modélisation spécifiques afin de mener différentes études de sûreté de fonctionnement.

MéDISIS 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 assistés par des outils et répertoires formant un EDS 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 (c.-à-d. 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.

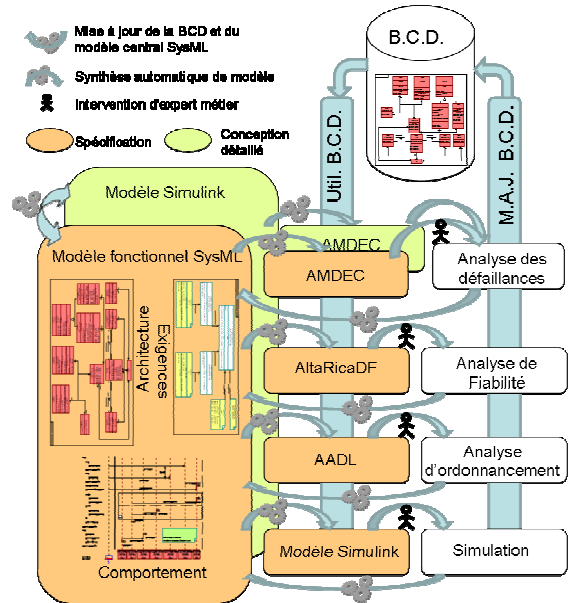


Figure 5 : Schéma de l'ensemble des processus MéDISIS

MéDISIS considère comme langage central le langage de modélisation système SysML [6,7]. En effet, celui-ci, intrinsèquement, profite aux points 1 et 3 puisqu'il permet une modélisation multi-vue qui s'adapte aux attentes des différents acteurs intervenant 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 contribuant 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 métamodèle 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, créées et réutilisées ; 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], les processus de traduction de SysML vers AADL et Simulink sont décrits.

## 5. 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, le véhicule LEA. Notre équipe a en charge le déploiement d'une méthode d'ingénierie de 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 MéDISIS est utilisée enrichie de nouveaux processus (c.-à-d. 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, certaines

fonctionnalités doivent être garanties, telles que les autotests, la fonction de détection du largage, les consignes moteur, la détection de fin de mission et la télémétrie. L'intérêt de rapprocher l'IS et les études de sûreté de fonctionnement a été constaté, dès les premières étapes du projet. La démarche suivie jusqu'à la conception préliminaire est résumée et les avantages constatés sont soulignés.

1- *Obtention des connaissances projets à partir des spécifications techniques du projet.* Cette étape suit un processus d'ingénierie classique supporté par un atelier SysML comprenant : la formalisation des exigences (act2) (diagramme d'exigences), la classification des besoins (act1) (diagramme de cas d'utilisation), la synthèse des spécifications techniques du partenaire, par cas d'utilisation (diagramme de séquence). La constitution des diagrammes paramétriques qualifiant les contraintes environnementales et techniques permet la modélisation des paramètres physiques dimensionnant du système (act3). Cette étape se termine par la définition de la vue organique du système (act4) et l'allocation des exigences aux différentes vues du modèle (Figure 6).



Figure 6 : Diagramme interne de bloc du véhicule LEA

2- *Analyse système.* L'étape d'analyse système est réalisée, l'apport principal de SysML [1] étant de réifier les analyses (act2, act3) par le biais de diagrammes paramétriques et l'établissement de liens (c.-à-d. *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 risque débute par la synthèse d'une pré-AMDEC (processus AMDEC, Figure 7). Le projet étant nouveau, les modes de défaillances génériques sont, dans un premier temps appliqués. À mesure de l'avancement de l'analyse, les résultats sont introduits dans la

BCD pour une définition plus précise 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 et les contraintes du système (modèle), identifiées comme étant impactées par des modes de défaillance (c.-à-d diagramme d'exigences et diagrammes paramétriques impactés, Figure 7).

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

Figure 7 : Extrait de l'AMDEC du système LEA

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 fonctionnelle sont classiquement résolus à ce stade [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]. La figure 8 montre le diagramme paramétrique modélisant un des résultats de cette étape.

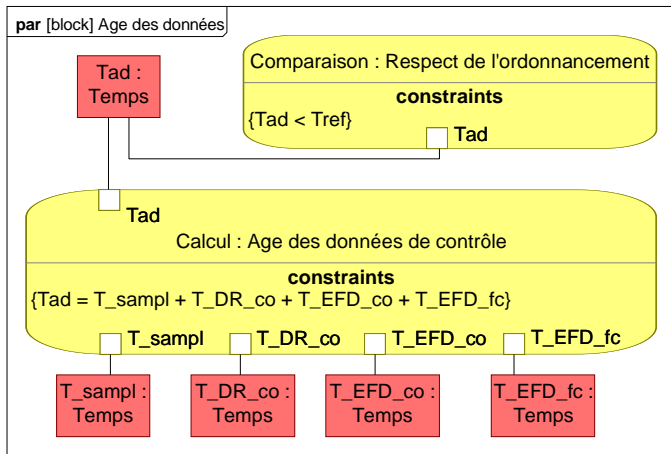


Figure 8 : Diagramme paramétrique décrivant la formule d'évaluation de l'âge des données.

6- *Injection des modèles de défaillance et jeux de tests*

Les similitudes relevées par [9] entre SysML et Simulink, permettent de définir la passerelle vers le domaine de la conception détaillée, donc la base du modèle de conception. C'est également à ce stade que les modèles des modes de défaillances les plus critiques sont injectés dans le modèle du système afin de réaliser les simulations du comportement du système et valider les choix de conception vis-à-vis des exigences de sûreté. En plus des jeux de tests fonctionnels issus des diagrammes de séquence (Figure 9) et des jeux de tests de charge issus des diagrammes paramétriques (Figure 8), sont donc introduits (niveau modèle) des jeux de tests dysfonctionnels. Ainsi, la conception est itérée, en prenant en compte des lois de contrôle robuste et leurs impacts sur les performances du système et leur effet sur les risques système sont estimés rapidement.

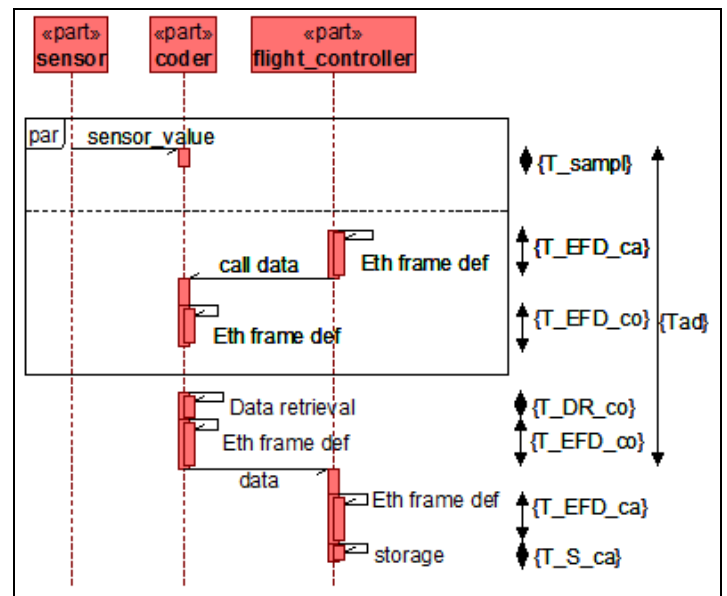


Figure 9 : Diagramme de séquence de la connexion Ethernet entre le calculateur et le module d'acquisition

6. CONCLUSION ET PERSPECTIVES

Cet article décrit comment la modélisation élémentaire des processus d'IS et de SdF permet d'identifier les besoins génériques nécessaires à un rapprochement efficace de leurs activités respectives. L'importance du rôle central de la BCD a été soulignée ; en illustrant notamment comment SysML permet de thésauriser le retour d'analyse des processus métier à travers les diagrammes paramétriques et les diagrammes interne de bloc. MéDISIS s'intègre efficacement dans une

stratégie d'ingénierie dirigée par les modèles, nous devons quantifier les gains en temps de conception actuellement qualifiés. Nos travaux s'orientent également vers la particularisation de MéDISIS pour les systèmes à COTS et la définition d'outils afin de mieux automatiser l'usage des diagrammes paramétriques dans nos processus.

#### RÉFÉRENCES

- [1] B. Cole, C. Delp & K. Donahue : *Piloting model-based engineering techniques for spacecraft concepts in early formulation* ; California Institute of Technology, publié par 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* ; 1<sup>st</sup> M-BED Workshop, DATE 2010, Dresde, Allemagne, 12 mars 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, n° 4, avril 2010, pp. 431-450.
- [4] J. Estefan. : *Survey of model-based systems engineering (MBSE) methodologies, Rev. B* ; INCOSE MBSE Initiative, 23 mai 2008.
- [5] F. Falempin et L. Serre : *French flight testing Program LEA Status in 2009* ; 16<sup>th</sup> AIAA/DLR/DGLR International Space Planes and Hypersonic Systems and Technologies Conference, Brême, Allemagne, 19-22 octobre 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.*, 1<sup>er</sup> novembre 2008.
- [8] OMG 2009 : *Unified Modeling Language* ; OMG Specification UML 2.2 Superstructure & UML 2.2 Infrastructure, 2009.
- [9] R. Snyder, D. Bocktaels et X. Feigntaels : *Validation fonctionnelle à l'aide d'une transformation SysML/Simulink* ; présenté lors des Journées Neptune 2010, Toulouse, 18-19 mai 2010, Génie Logiciel, n° 93, juin 2010, pp. 49-53.