Utilisation du MBSA pour l'évaluation d'un événement sécuritaire ferroviaires SIL2

Model-Based Safety Approach for SIL2 railway safety event assessment

DUTHOIT Sébastien
Groupe LGM
Villeneuve d'ascq (FR) / Charleroi (BE)
sebastien.duthoit@lgm.eu

TBC
ALSTOM
Charleroi (BE)
TBC

RESUMÉ — L'approche par les modèles tend à prendre une part importante dans les analyses de sureté de fonctionnement afin de répondre à la complexité croissante des systèmes étudiés. Le contexte normatif Ferroviaire demande une grande maitrise des approches afin de répondre aux besoins de certification. L'approche par les modèles n'y est pas encore déployée à grande échelle. L'objectif de cette publication est de présenter un cas pratique d'une modélisation utilisée dans le cadre de la démonstration d'un niveau d'intégrité SIL2 pour une fonction sécuritaire et en modélisant les tests périodiques (tests automatiques ou en maintenance préventive). Elle présentera également les enjeux de l'utilisation de cette méthodologie dans le contexte normatif ferroviaire.

ABSTRACT— The model-based approach tends to play an increasingly role in functional safety analyses to respond to the growing complexity of studied systems. The railway certification context requires a high mastery of approaches to meet certification needs. The model-based approach is not yet widely deployed in this context. The objective of this publication is to present a practical case of a modeling used in the context of an SIL2 integrity level safety function demonstration and by modeling periodic tests (automatic tests or preventive maintenance). The use of this methodology in the railway regulatory context will also be presented.

MOTS-CLEFS, KEYWORDS — MBSA, Sécurité, Safety, Ferroviaire, Railways, Tests Périodiques, Periodic Tests EN50126, EN50129,

I. INTRODUCTION

Les systèmes ferroviaires intègrent de nombreuses fonctions de sécurité essentielles. Les normes régissant ce secteur incluent les normes CENELEC EN50126-1[R2] et EN50126-2[R3] pour la gestion de la sécurité tout au long du cycle de vie, EN50128[R4] et EN50657[R5] pour la gestion de la sécurité logicielle, ainsi que la norme EN50129[R6] pour la gestion de la sécurité matérielle. Dans leurs versions les plus récentes, ces normes définissent des niveaux d'intégrité de sécurité allant du niveau "Basic Integrity", le moins exigeant, au niveau SIL4, le plus exigeant.

Alstom est acteur clé pour la conception et la fourniture de systèmes ferroviaires. Le groupe LGM supporte Alstom dans les analyses de sureté de fonctionnement.

Face à la complexité croissante des systèmes ferroviaires, les analyses basées sur une approche traditionnelle deviennent de plus en plus complexes à réaliser. L'approche par modélisation se présente alors comme une solution pertinente pour maîtriser efficacement les scénarios critiques et gagner en efficacité dans les analyses.

L'objectif de cet article est de présenter un cas d'étude illustrant l'utilisation d'une modélisation pour quantifier un événement redouté visant à atteindre un niveau SIL2. Dans une première partie, nous présenterons le cas d'étude. Dans une seconde partie, nous aborderons les différentes approches employées et les résultats obtenus. Enfin, dans une troisième partie, nous expliquerons

comment cette approche peut être intégrée dans le processus de certification. Nous conclurons par une synthèse des principaux enseignements tirés de cette étude.

II. DESCRIPTION DU CAS D'ÉTUDE

Le contrôle des moteurs d'un train s'effectue à travers l'utilisation d'un coffre de traction. Ce dernier se compose d'une partie de contrôle et d'une partie de puissance chargée de convertir l'énergie de la caténaire vers les moteurs. Un déplacement intempestif du train représente un événement sécuritaire identifié lors de l'analyse préliminaire des risques.

Afin d'atteindre le niveau de sécurité approprié au niveau système, des fonctions indépendantes sont définies et allouées à plusieurs composants du train, chacune avec son niveau d'intégrité (niveau « SIL » pour couvrir les pannes systématiques) et son taux de panne (« TFFR » pour couvrir les pannes aléatoires).

Le coffre de traction doit être capable d'inhiber en toute sécurité les impulsions de contrôle du moteur afin d'éviter toute application d'un effort de traction intempestif, que ce soit en station ou lors d'un freinage d'urgence. Un niveau SIL2 est requis sur cette fonction.

Contrairement à la norme IEC 61508 [R8], les normes CENELEC ne nécessitent pas de diversification fonctionnelle et technologique. Cependant, cette diversification peut tout de même être mise en œuvre afin de faciliter l'atteinte du niveau de sécurité requis tout en éliminant les modes de défaillance communs.

Pour l'inhibition des impulsions et afin d'assurer une diversification technologique, cette fonction est réalisée à la fois par une fonction logicielle et une fonction matérielle. Pour garantir la diversification fonctionnelle, deux interruptions sont effectuées : une interruption des impulsions elles-mêmes et une interruption de l'alimentation du circuit de commande des grilles des transistors de puissance de l'onduleur.

Deux commandes indépendantes arrivent du niveau train : une première SIL2 liée à la coupure des impulsions elle-même et une seconde SIL4 liée à l'application d'un freinage d'urgence.

L'implémentation de la fonction peut être résumée par le diagramme suivant :

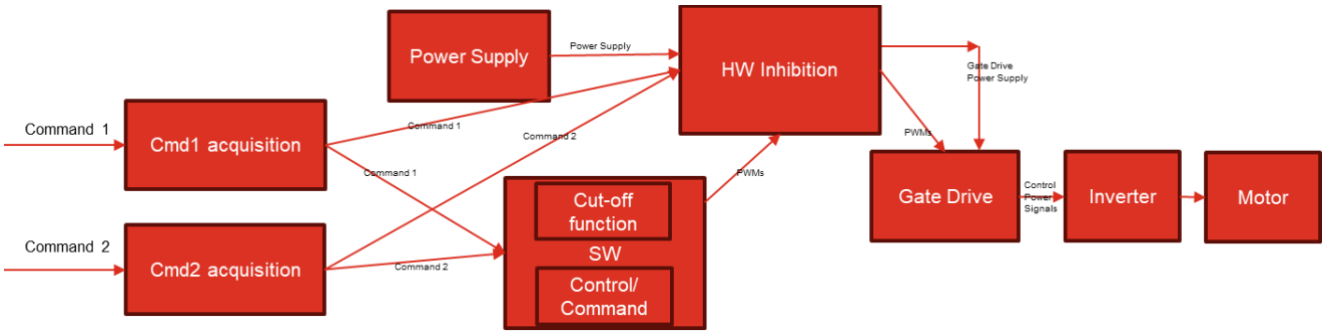

Fig. 1. Description de la fonction de coupure des impulsions d'un coffre de traction.

III. DESCRIPTION DE L'APPROCHE ET RESULTATS

Afin de modéliser cette fonction, deux solutions technologiques ont été identifiées :

- Le logiciel System Analyst,
- Le logiciel OpenAltaRica,

L'objectif dans les deux cas est de quantifier l'événement suivant : Non-inhibition des impulsions lors d'une commande d'inhibition,

1) Modélisation avec System Analyst

À l'aide de System Analyst, le cas d'étude a été représenté par les blocs suivants :

- Un bloc d'acquisition qui englobe les acquisitions des deux commandes et de l'alimentation du circuit de commande de grille,
- Un bloc logiciel,
- Un bloc matériel,
- Le circuit de commande de grille,

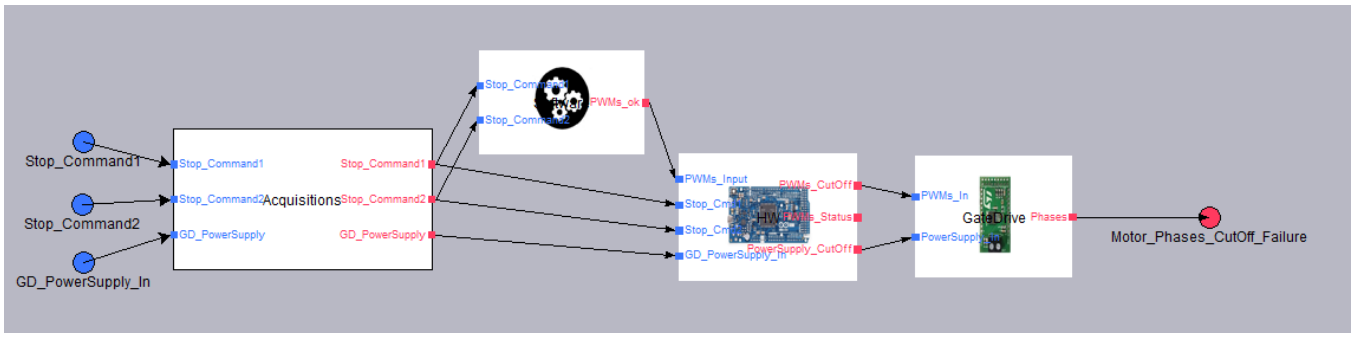

Fig. 2. Modélisation du cas d'étude sous System Analyst.

Les entrées logiques sont conçues selon une approche dite "fail-safe", où l'état haut représente une autorisation et l'état bas une inhibition. Cette méthode permet de sécuriser les pertes d'alimentation et les coupures de câblage. Dans le bloc d'acquisition, le mode de panne susceptible de causer une non-inhibition des impulsions est un maintien d'une entrée à l'état haut qui sera modélisé pour chacune des deux entrées.

Les composants des systèmes doivent être testés régulièrement pour garantir qu'ils sont exempts de défaillances. Par conséquent, une loi périodique simple est mise en œuvre en se basant sur un taux de panne intrinsèque du composant et un temps de latence correspondant au temps maximal nécessaire pour détecter toute panne du composant.

Les acquisitions étant de niveau SIL2 et conformément à la norme EN50129 [R6], leurs taux de panne aléatoire doivent être strictement inférieurs à 1.00E-6 défaillances par heure. Une valeur de 9.00E-6 défaillances par heure est prise pour tout élément SIL2 afin de faciliter l'analyse.

Étant donné que l'application de la coupure des impulsions est réalisée en station, il est possible de détecter une défaillance sur cette entrée à plusieurs reprises au cours une journée de service. En revanche, comme l'activation d'un freinage d'urgence n'est pas systématique, un test au démarrage est instauré, soit une durée de 24 heures. Entre chaque station, un pire cas d'une durée d'une heure est pris en compte. Les deux modes de pannes liés aux blocs d'acquisition ont donc les caractéristiques suivantes :

- Entrée de coupure des impulsions (commande 1) : Loi simple de tests périodiques avec un taux de panne de 9.00E-6 défaillances par heure et un temps de latence de 1 heure,
- Entrée de freinage d'urgence (commande 2) : Loi simple de tests périodiques avec un taux de panne de 9.00E-6 défaillances par heure et un temps de latence de 24 heures,

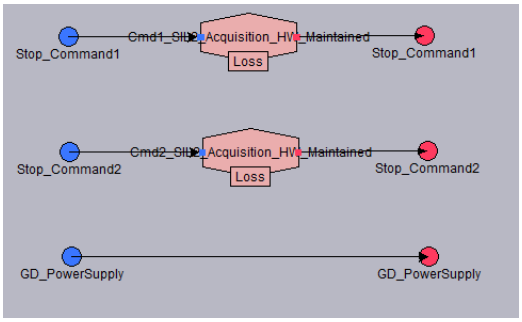

Fig. 3. Bloc Acquistions

Pour le bloc logiciel, celui-ci n'est pas pris en inclus dans une approche quantitative étant donné qu'il n'est pas soumis aux pannes aléatoires. Il est simplement modélisé par sa logique fonctionnelle qu'il gère à travers ses deux entrées :

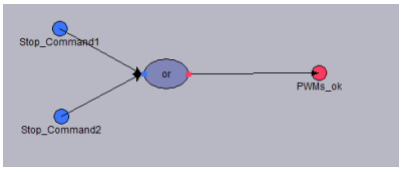

Fig. 4. Bloc de coupre logiciel

Pour le bloc de coupure matériel, il est composé de deux modes de pannes potentiels :

- Une absence de coupure des impulsions due à un mécanisme de coupure matérielle défaillant,
- Une absence de coupure de l'alimentation du circuit de commande des grilles,

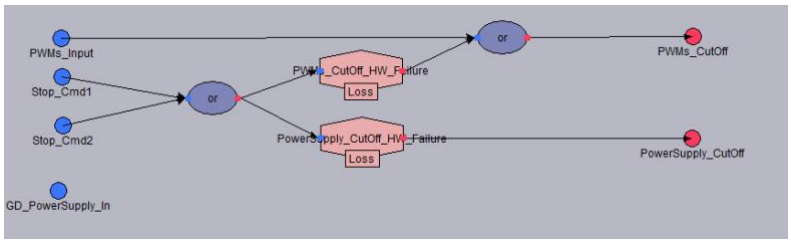

Fig. 5. Bloc de coupure matériel

Les deux modes de pannes associés au bloc de coupure matérielle présentent les caractéristiques suivantes :

- Coupure des impulsions : Loi simple de tests périodiques avec un taux de panne de 9.00E-6 défaillances par heure et un temps de latence de 24 heure,
- Coupure de l'alimentation : Loi simple de tests périodiques avec un taux de panne de 9.00E-6 défaillances par heure et un temps de latence de 24 heures,

Enfin pour le bloc représentant le circuit de commande des grilles, la logique fonctionnelle suivante a été modélisée :

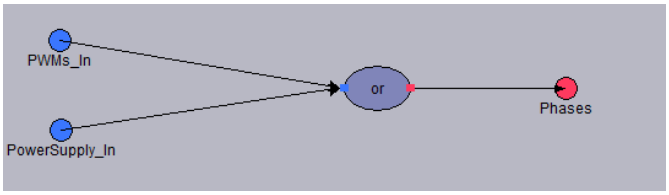

Fig. 6. Bloc du circuit de commande des grilles

Une simulation interactive ainsi qu'un arbre de défaillance peut être généré afin de calculer une probabilité conditionnelle (CFI) :

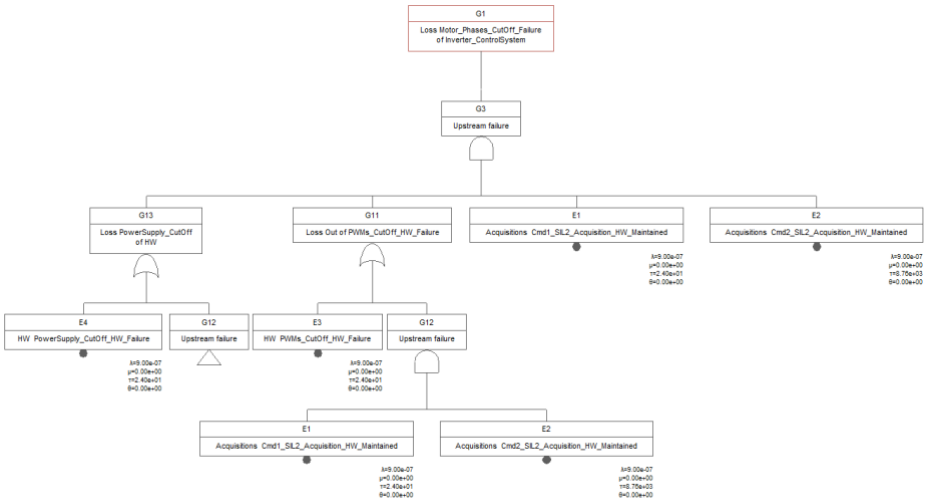

Fig. 7. Arbre de défaillance généré à partir de System Analyst

1) Modélisation avec OpenALtaRica

L'outil OpenAltaRica nécessite une maîtrise directe du langage AltaRica 3.0 [R8]. En effet, il est encore dépourvu de représentation graphique. La même décomposition des blocs a été réalisée. Chacun de bloc ont été modélisé avec les points suivants :

- PTCMode:
- État du composant pour représenter si le composant est en opération (OPERATION) ou en test (TEST_MAINTENANCE),
- Modes de panne / Taux de panne : Définition des modes de pannes et des taux associés,
- Variables de flux,
- Événements,

- Transition associée aux événements,
- Assertions

Voici un exemple d'un bloc d'acquisition modélisé dans le langage :

```
// Name: Cmd_Tested_Acquisition_HW
// Description: Logical input acquisition
class Cmd_SIL2_Acquisition_HW
    // Global Parameters
    PTCMode _Mode (init=OPERATION);

    // Local Parameters
    parameter Real lambda_failure_HIGH = 9E-7;
    CState _State_HIGH (init=WORKING);
    parameter Real lambda_failure_LOW = 1e-5;
    CState _State_LOW (init=WORKING);


    // Variables - Flows
    Logic_Signal in, out (reset=HIGH);

    // Events
    event failure_HIGH (delay=exponential(lambda_failure_HIGH));
    event failure_LOW (delay=exponential(lambda_failure_LOW));

    // Transitions
    transition failure_HIGH: ( _State_HIGH == WORKING and _State_LOW == WORKING) -> _State_HIGH := FAILED;
    transition failure_LOW: (_State_HIGH == WORKING and _State_LOW == WORKING) -> _State_LOW := FAILED;

    // Assertions
    assertion out:= if _State_HIGH == FAILED then HIGH else if _State_LOW == FAILED then LOW else in;

end
```

Fig. 8. Modélisation d'un bloc composant en AltaRica 3.0

Pour gérer l'aspect des tests périodiques, l'approche dans la modélisation des combinaisons de maintenance en AltaRica 3.0, décrite dans [R1] a été mise en œuvre. L'objectif est de prendre en compte les éléments de testabilité de manière indépendante par rapport à l'architecture et aux composants eux-mêmes. En effet, les tests peuvent être déployés uniquement au niveau fonctionnel. Ainsi, c'est à travers un bloc de supervision que ces tests sont gérés. Ce bloc est constitué uniquement de transitions pour représenter les changements entre les différents modes opérationnels :

```
block Supervisor_Cmd1_SIL2_Acquisition_HW_HIGH
    //Components
    embeds main.Architecture.myCmd1_SIL2_Acquisition_HW as my_Item;
    //Parameters
    PTCMode _Mode (init=OPERATION);
    parameter Real latency_time = 1;
    parameter Real test_maintenance_time = 0;
    parameter Real pSTM = latency_time;
    parameter Real pCTM = test_maintenance_time;
    //Events
    event startTestMaintenance (delay=Dirac(pSTM));
    event completeTestMaintenance (delay=Dirac(pCTM));
    // Transitions
    transition startTestMaintenance: _Mode == OPERATION -> {_Mode := TEST_MAINTENANCE; my_Item._Mode := TEST_MAINTENANCE;}
    transition completeTestMaintenance: _Mode == TEST_MAINTENANCE -> {
                    _Mode := OPERATION;
                    my_Item._Mode := OPERATION;
                    my_Item._State_HIGH := switch {
                        case my_Item._State_HIGH == FAILED : WORKING
                        default: WORKING};
                    }
end
```

Fig. 9. Modélisation d'un bloc de supervision en AltaRica 3.0

Enfin, un bloc d'architecture modélise l'ensemble de l'architecture du cas d'application et permet de relier les différents composants pris en compte ainsi que les logiques fonctionnelles, telles que la logique du logiciel. Ce bloc est constitué uniquement d'assertions, car il représente simplement l'intégration des sous-composants entre eux.

```
block Architecture

    // System inputs
    Logic_Signal Stop_Command1 (reset=HIGH);
    Logic_Signal Stop_Command2 (reset=HIGH);
    Power_Supply GD_PowerSupply_In (reset=ACTIVE);

    // System outputs
    PWMs Motor_Phases_Cutoff_Failure (reset=ACTIVE);

    // Internal
    PWMs Motor_Phases_SW (reset=ACTIVE);
    Logic_Signal HW_CMD_AND (reset=HIGH);
    Logic_Signal HW_PWMs_Cut (reset=HIGH);
    Logic_Signal HW_PS_Cut (reset=HIGH);
    PWMs Motor_Phases_HW (reset=ACTIVE);
    PWMs PS_HW (reset=ACTIVE);

    //Observation
    Boolean Safety (reset=false);
    Boolean Availability (reset=false);

    //Components
    Input_Controler my_Input_Controler;
    Cmd_SIL2_Acquisition_HW myCmd1_SIL2_Acquisition_HW;
    Cmd_SIL2_Acquisition_HW myCmd2_SIL2_Acquisition_HW;
    PowerSupply myPowerSupply_Acquisition;
    PowerSupply_CutOff_HW myPowerSupply_CutOff_HW;
    PWMs_CutOff_HW myPWMs_CutOff_HW;


    assertion
        //Acquisitions
        my_Input_Controler.in1 := Stop_Command1;
        my_Input_Controler.in2 := Stop_Command2;
        myCmd1_SIL2_Acquisition_HW.in := my_Input_Controler.out1;
        myCmd2_SIL2_Acquisition_HW.in := my_Input_Controler.out2;
        myPowerSupply_Acquisition.in :=GD_PowerSupply_In;

        //Software
        Motor_Phases_SW := if (myCmd1_SIL2_Acquisition_HW.out == LOW or myCmd2_SIL2_Acquisition_HW.out == LOW) then STOPPED else ACTIVE;

        //Hardware
        HW_CMD_AND := if (myCmd1_SIL2_Acquisition_HW.out == LOW or myCmd2_SIL2_Acquisition_HW.out == LOW) then LOW else HIGH;
        myPowerSupply_CutOff_HW.in := HW_CMD_AND;
        myPWMs_CutOff_HW.in := HW_CMD_AND;

       HW_PWMs_Cut := myPWMs_CutOff_HW.out;
       HW_PS_Cut := myPowerSupply_CutOff_HW.out;
       Motor_Phases_HW := if (Motor_Phases_SW == STOPPED or HW_PWMs_Cut == LOW) then STOPPED else ACTIVE;
       PS_HW := if (myPowerSupply_Acquisition.out == LOSS or HW_PS_Cut == LOW) then STOPPED else ACTIVE;

       // Gate Drive
       Motor_Phases_Cutoff_Failure := if (Motor_Phases_HW == STOPPED or PS_HW == STOPPED) then STOPPED else ACTIVE;

       //Observation
       Safety := if ((my_Input_Controler.out1 == LOW or my_Input_Controler.out2 == LOW) and Motor_Phases_Cutoff_Failure == ACTIVE) then true else false;

    observer Boolean Observer_Safety = Safety;
end
```

Fig. 10. Modélisation du bloc d'architecture

Un observateur de sécurité est déclenché quand les impulsions ne sont pas désactivées tout en ayant les deux demandes de demande d'arrêt désactivâtes. Une simulation interactive, ainsi qu'une simulation stochastique sur une année avec un échantillon de 10 000 exécutions, a été réalisée pour quantifier cet événement.

IV. L'APPROCHE PAR LES MODELES DANS LE FERROVIAIRE

Le cas d'usage décrit dans le paragraphe précédent démontre la faisabilité d'utiliser l'approche par les modèles pour quantifier des événements de sécurité dans le domaine ferroviaire. Le management de la sécurité, tel qu'il est défini dans le dossier de sécurité et dont la structure est décrite dans l'EN50126-2[R3] et l'EN50129[R6] restera d'application.

L'approche par les modèles peut être utile pour :

- Faciliter les interactions entre les équipes d'ingénierie et les équipes de sûreté de fonctionnement. En effet, il est souvent complexe de réviser les arbres de défaillance avec des concepteurs peu familiarisés avec une logique de dysfonctionnement.
- Favoriser une approche basée sur la réutilisation de blocs de base au sein d'une architecture pour un projet.
- Simplifier la gestion des modes opérationnels, qui peuvent être multiples dans les applications ferroviaires, en utilisant des superviseurs capables d'ajuster les effets en fonction du mode actif.

L'approche par modélisation nécessite l'utilisation d'outils spécifiques, qui ne sont pas encore pleinement reconnus. Les normes EN50128[R4]/EN50657[R5] et l'EN50129[R6] exigent une évaluation approfondie des outils employés dans le cadre du développement pour garantir qu'ils ne peuvent pas avoir d'impact direct sur la sécurité. Les outils utilisés dans cette étude n'ont pas encore été reconnus ni analysés.

Pour pouvoir adopter l'approche par modélisation sans avoir à certifier les outils, il est nécessaire d'effectuer une revue des résultats de l'analyse par un vérificateur, directement au niveau des arbres de défaillance générés, plutôt qu'au niveau du modèle lui-même.

V. Conclusion

Ce cas d'application montre que l'approche par les modèles peut être parfaitement intégrée dans une démonstration de sécurité ferroviaire. Bien que ce cas soit relativement simple, il sera nécessaire à l'avenir de modéliser des systèmes plus complexes, composés de configurations multiples et de modes opérationnels variés.

Si l'approche par les modèles et les outils pour les réaliser ainsi que la vérification et la validation du modèle venait à augmenter en confiance, le secteur pourrait tout à fait l'intégrer directement dans les certifications auprès des évaluateurs indépendants.

Les synergies entre les équipes d'ingénierie et les équipements de sûreté de fonctionnement, que ce soit pour enrichir un modèle d'ingénierie ou pour assurer la cohérence entre deux modèles, ne sont actuellement pas encore suffisamment développées pour garantir une intégration réussie de l'approche par modélisation dans le cycle de vie d'un produit ferroviaire.

VI. Acronymes

HW : HardWare

SW : SoftWare

TFFR : Tolerable Failure Rate

SIL : Safety Integrity Level

MBSA: Model-Based Saefty Assessment

PBIT: Power-on Build-In-Test

CFI: Conditional Failure Intensity

VII. References

[R1] *Modélisation de combinaisons de maintenance en AltaRica 3.0*, Michel Batteux, Tatiana Prosvirnova, Antoine Rauzy, HAL Id: hal-03462797

[R2] EN 50126-1. *Applications ferroviaires - Spécification et démonstration de la fiabilité, de la disponibilité, de la maintenabilité et de la sécurité (FDMS) - Partie 1 : processus FMDS générique*

[R3] EN 50126-2. *Applications ferroviaires - Spécification et démonstration de la fiabilité, de la disponibilité, de la maintenabilité et de la sécurité (FDMS) - Partie 2 : approche systématique pour la sécurité*

[R4] EN 50128. *Applications ferroviaires - Systèmes de signalisation, de télécommunication et de traitement - Logiciels pour systèmes de commande et de protection ferroviaire*

[R5] EN 50657. *Applications ferroviaires - Applications du matériel roulant - Logiciels embarqués*

[R6] EN 50129. *Applications ferroviaires - Systèmes de signalisation, de télécommunications et de traitement - Systèmes électroniques de sécurité pour la signalisation*

[R7] IEC61508 *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité*

[R8] AltaRica 3.0 - Language Specification