Double cœur et preuve formelle pour automatismes SIL4

Conference Paper · December 2016

DOI: 10.4267/2042/61819

CITATIONS
7

READS
220

1 author:

Thierry Lecomte
ClearSy System Engineering
54 PUBLICATIONS **418** CITATIONS

SEE PROFILE

Bi-processeur et preuve formelle pour automatismes SIL4
Bi-processor and formal proof for SIL4 automatisms

Thierry LECOMTE
ClearSy
320, avenue Archimède, Les Pléiades3, bat A
13857 Aix en Provence France
thierry.lecomte@clearsy.com

Résumé
L'architecture sécurisée double cœur combine des techniques logicielles et matérielles compatibles EN5012{6, 8, 9}, et permet de construire des automatismes sécuritaires SIL4 à bas coût. Cette architecture combine astucieusement méthode formelle de développement et plateforme d'exécution sécurisée simple. Cette architecture a été mise en œuvre dans le cadre d'applications industrielles certifiées et fait l'objet d'un projet de recherche collaboratif afin d'en faire un produit générique.

Summary
The bi-processor secure architecture combine software and hardware techniques, compatible with EN5012{6, 8, 9}, and allows to design low cost, high integrity automatisms. This smart architecture combines formal software development with proof and simple, secure execution platform. This architecture has been applied to several industrial settings and is being improved and made generic through a collaborative research and development project.

Objectifs

La communication présente une architecture sécurisée combinant des techniques logicielles et matérielles compatibles EN5012{6, 8, 9}, et permettant de construire des automatismes sécuritaires SIL4 à bas coût. Des exemples d'applications industrielles sont détaillés.

Contexte

Le développement d'applications sécuritaires implique

- des coûts de développement plus importants: un logiciel SIL coûte 2 fois plus cher qu'un développement traditionnel (équipe de sureté de fonctionnement (safety) dédiée, vérification et validation (V&V) supplémentaires, redondance applicative, etc.)
- des plateformes d'exécution plus onéreuses : le hardware est spécifique et de diffusion plus confidentielle, ou pour les niveaux SIL3 et SIL4, les automates existants sont chers et parfois inadaptés à un environnement ferroviaire (aux normes de l'embarqué par exemple)
- des contraintes de mise en œuvre: les environnements de travail sont souvent imposés pour conserver le tampon «SILx» de l'équipement

Les implications sont multiples: le développement de telles applications n'est pas neutre pour l'entreprise. Les solutions proposées sont rigides. Le coût élevé de l'application peut empêcher son implémentation. Le personnel n'est pas aguerri à des développements de ce type, et la sous-traitance de certaines activités est difficile. Enfin d'un point de vue sociétal, des moyens de sécurisation de la population peuvent ne pas être mis en œuvre pour des raisons économiques.

Méthode

Une architecture bi-processeur a été conçue afin de répondre à ces problèmes. Il s'agit d'une plateforme matérielle/logicielle (voir figure 1) articulée autour d'un hardware standard à bas coût et d'un processus de développement logiciel prouvé et redondé, permettant de développer des automatismes sécuritaires SIL4 à bas coût.

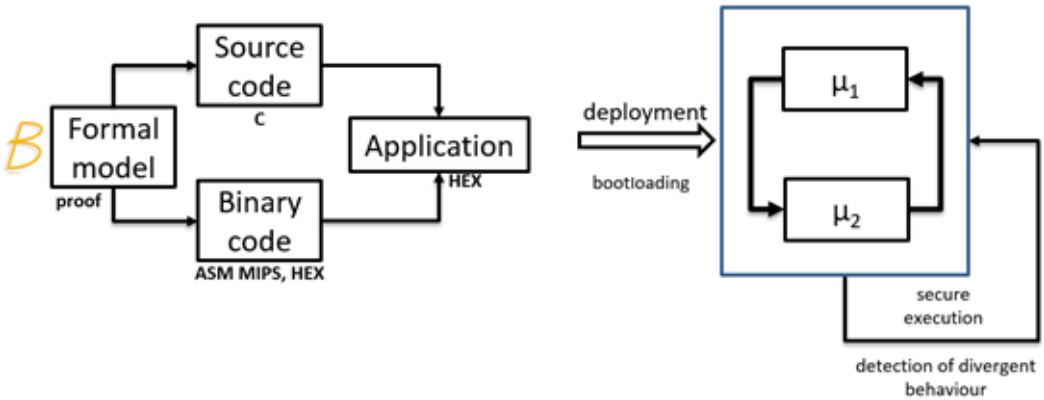

Figure 1. Chaine de développement formelle et plateforme d'exécution

Le processus de développement consiste à spécifier le modèle d'un programme à algorithmique bornée, prouver que ce modèle vérifie certaines propriétés définies par l'utilisateur, et générer automatiquement le code binaire redondant avec *a minima* preuve automatique d'absence d'erreur de programmation (par exemple débordement mémoire, division par zéro, accès tableau hors domaine, etc.).

La plateforme d'exécution est basée sur deux calculateurs PIC 32 (microcontrôleurs 32 bits). Chaque microcontrôleur va exécuter en séquence deux instances diversifiées du même programme. Les principes de sécurisation reposent sur la détection de divergence de comportement entre instances s'exécutant sur un même microcontrôleur et entre les deux microcontrôleurs. Les principes de sécurisation sont non-accessibles au développeur qui ne peut donc pas les altérer.
Cette plateforme propose une puissance de calcul utile de 20 à 100 MIPS. En effet, les 200 MIPS crête du PIC 32 le plus performant sont divisés par un facteur un peu supérieur à 2 du fait que sur un microcontrôleur deux instances du même programme doivent être exécutées en séquence, les résultats entre instances et entre microcontrôleurs doivent être comparés, et la cohérence mémoire doit être vérifiée. La plateforme est par ailleurs capable de détecter l'impossibilité de contrôle des sorties. Toute anomalie de comportement entraine alors un redémarrage voire un effacement de la mémoire et l'entrée dans une boucle sans fin (mode défaut). Pour améliorer la disponibilité du dispositif, il est alors possible d'utiliser deux bi-processeurs en parallèle.

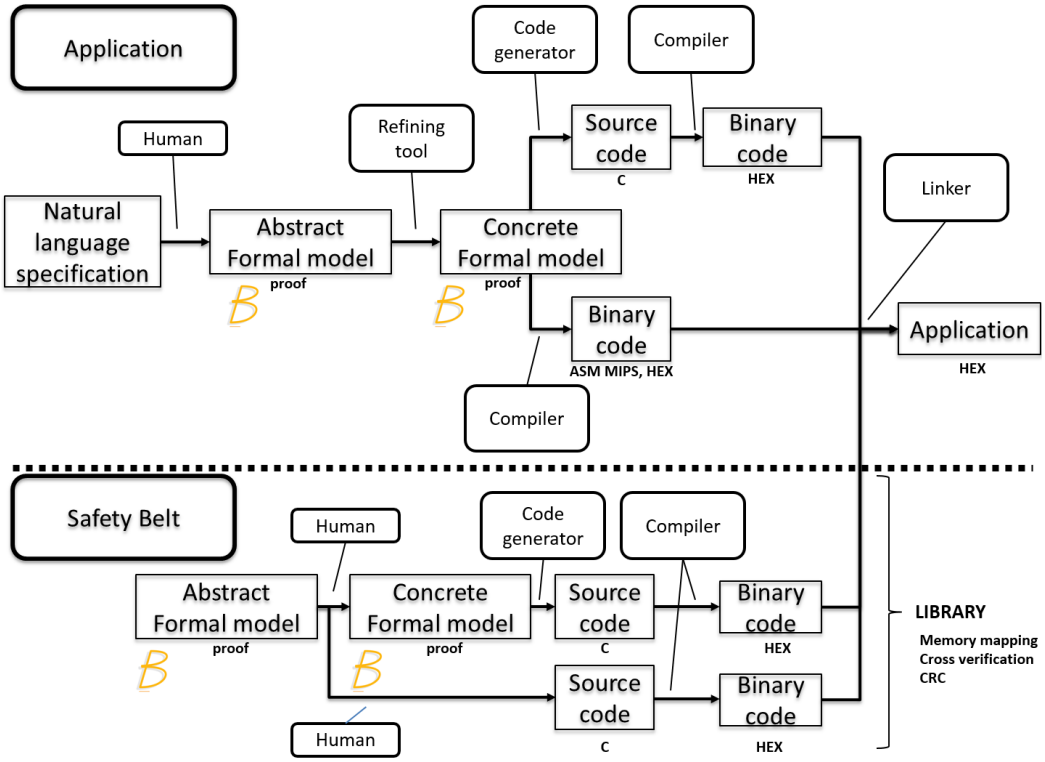

Figure 2. Processus de développement d'une application et de la librairie de fonctions assurant la sécurité logicielle de l'architecture

La chaîne de développement de logiciels est formelle : elle autorise la spécification et la conception formelle d'un logiciel exprimé sous la forme de modèles B. Les logiciels visés sont des logiciels cycliques pour lesquels aucune interruption ne vient en modifier les variables d'état (il est par contre possible que l'acquisition des entrées soit réalisée au travers d'interruptions). Sont supportées l'arithmétique entière et la logique booléenne pour la représentation par exemple de machines à états, d'équations booléennes d'enclenchements, etc. Les modèles B sont prouvés non contradictoires et conformes à la spécification initiale. Le modèle B, unique, représentant l'implémentation du logiciel est ensuite utilisé par une double chaine de génération de code pour produire 1) un code C qui sera compilé au format binaire HEX par le compilateur standard Microchip 2) un code HEX grâce à un compilateur B/ASM MIPS/HEX. Le binaire final, constitué des deux instances produites et d'un code-librairie regroupant les fonctions de sécurisation (comparaison des résultats, contrôle d'intégrité, testeur d'instructions, etc.), est ensuite téléchargé à l'identique à bord de chaque PIC 32 au moyen d'un boot-loader.

La figure 2 présente l'enchainement des différentes actions à réaliser pour obtenir le code application (partie *application*) et la librairie de fonctions assurant la sécurité logicielle de l'architecture (partie *safety belt*).
Le code de l'application s'obtient au travers d'un processus en majeur partie outillé. L'étape d'obtention du modèle formel abstrait, réalisée manuellement, nécessite une vérification indépendante de conformité avec la spécification du besoin exprimée en langage naturel. La transformation du modèle abstrait en modèle implémentable est pour partie automatisée mais requiert encore des interactions expertes avec l'environnement de développement. La génération de code binaire repose d'une part sur le générateur de code C de l'Atelier B (C_4B) et le compilateur Microchip (C32), sur le compilateur B/Hex de ClearSy (b32) d'autre part.
Le code des librairies de fonctions sécuritaires repose sur un code C issu de modélisation formelle ou de machines de base (modèle formels dont l'implantation est développée manuellement et qui couvrent des fonctionnalités bas-niveau). Ce code C est ensuite compilé avec le compilateur Microchip (C32) avant d'être linké. Ce code implémente le séquencement des instances, les comparaisons internes et externes, l'intégrité des logiciels (CRC), le mapping mémoire, etc.

La plateforme ainsi obtenue permet de détecter les erreurs de compilation et d'exécution (divergence, problèmes HW, commande des sorties, chargement des applications) sans avoir besoin d'utiliser des générateurs de code certifiés, la démonstration de sécurité n'en montrant pas la nécessité. Les modes communs sont gérés au travers de plusieurs mécanismes comme par exemple le testeur d'instructions qui vérifie en permanence que les instructions de chaque microcontrôleur sont opérationnelles (l'instruction addition réalise bien une addition, etc.). Le cadre d'utilisation de cette plateforme, bien que contraint par les technologies utilisées, n'en est pas trop spécifique et peut être utilisé pour des applications légères.

Cette plateforme peut enfin se transposer facilement dans des domaines industriels ayant recours aux automatismes pour des fonctions liées à la sécurité.

Projet R&D collaboratif

Du fait des succès techniques rencontrés lors de la mise en œuvre de cette technologie sur des produits à l'export, un projet collaboratif (FUI21 LCHIP), porté par ClearSy, a été constitué avec pour objectif de rendre la technologie générique au travers d'une connexion amont avec des langages métiers (Domain Specific Languages) et d'une automatisation complète du processus formel (voir figure 3).

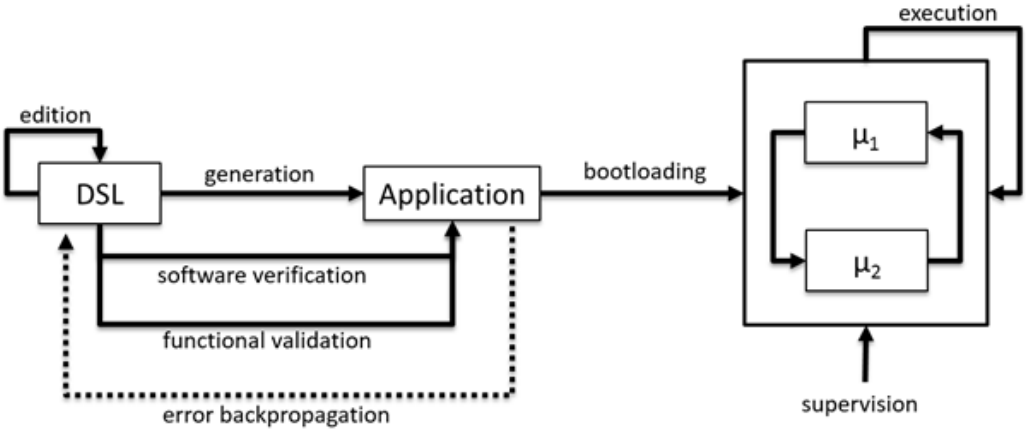

Figure 3. Schéma de principe des fonctionnalités prévues dans le cadre du projet LCHIP

La conjonction de ces deux avancées doit permettre à terme à un ingénieur de pouvoir produire plus facilement une application sécurisée, à algorithmique bornée, en utilisant son formalisme habituel tout en occultant les détails de l'étape de vérification et validation formelle.

Le projet LCHIP réutilise nombre de résultats obtenus lors de projets de R&D industriels ou collaboratifs passés pour adresser ces deux thématiques.
La traduction de schéma à relais en réseaux de Pétri (thèse IFSTTAR), la traduction bidirectionnelle UML/B (FP7 Rodin) et la validation d'applications de maintenance spécifiées en Grafcet/texte structuré (SNCF WinGUIDEP) doivent assurer la capacité à transformer des schémas à relais et des grafcets en B puis de pouvoir rétropropager les obligations de preuve B non prouvables (les erreurs) sur les modèles initiaux non formels.
L'outil de raffinement automatique BART (ANR Rimel), le générateur d'obligations de preuve traçable (ANR Cercles-2), la traduction des obligations de preuve B en Why3 et l'adaptation du prouveur Alt-Ergo (ANR Bware) doivent assurer la capacité à transformer automatiquement un modèle abstrait en modèle implémentable puis à le prouver.

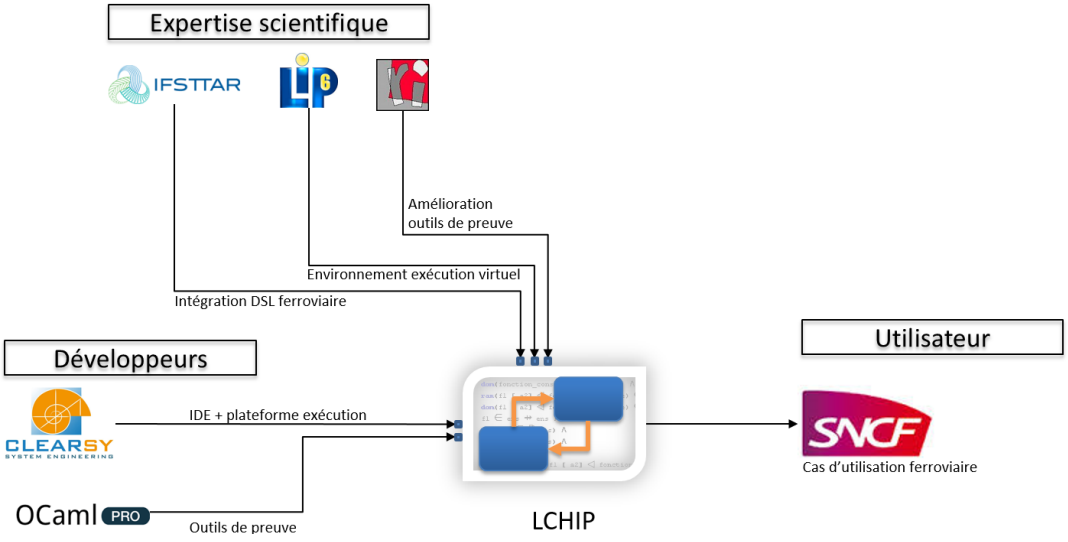

Figure 4. Le consortium de réalisation du projet de R&D collaboratif LCHIP

Le consortium de réalisation a été construit autour de ces deux thèmes (voir figure 4). Il contient :

- deux fournisseurs de technologies en charge de construire le cœur de la plateforme LCHIP,
- trois laboratoires assurant l'intégration de DSL ferroviaires, d'un environnement d'exécution virtuel et d'outils de preuve tiers.
- un utilisateur en charge d'évaluer la technologie sur un cas d'études réel et d'apprécier l'applicabilité de celle-ci à ses besoins (homologation).

Le projet, débuté en 2016, dure 3 années. Un extrait du planning de réalisation est donné figure 5. Trois dates clé sont à retenir :

- Q3 2017 : mise à disposition publique d'un starter kit contenant
  - une carte mère, comprenant des entrées/sorties, un processeur de maintenance et une connectique réseau,
  - une carte fille avec un bi-processeur à base de PIC32
  - l'environnement de développement en version 1
- Q2 2018 : l'environnement de développement en version 2
- Q2 2019 : l'environnement de développement en version 1

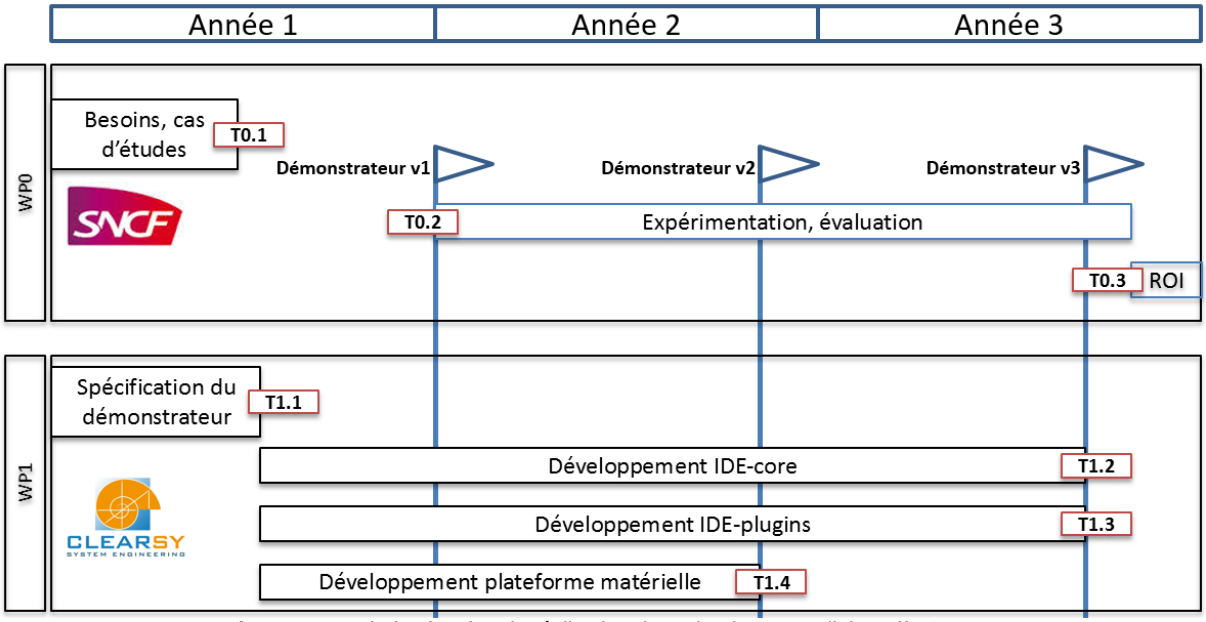

Figure 5. Extrait du planning de réalisation du projet de R&D collaboratif LCHIP

Conclusion

La plateforme bi-processeur SIL4 a été utilisée avec succès pour le développement du contrôle/commande des portes palières du métro Sao Paulo (ligne 15). Pour améliorer la compétitivité de la version initiale de ce contrôle/commande basé sur des automates Siemens S7 pour les lignes 2 et 3 développée en 2009, les fonctionnalités de gestion des entrées/sorties et les automatismes ont été répliqués à l'aide d'une carte électronique spécifique embarquant 8 microcontrôleurs PIC 3. Avec un volume divisé par 4, des coûts de développement sensiblement réduits et une certification SIL4 par CERTIFER en cours, cette plateforme a démontré tout son intérêt. Elle sera par ailleurs installée dans le métro de Stockholm dès 2017.

La plateforme bi-processeur SIL4 est utilisée pour la gestion des entrées/sorties de la passerelle SATURN SIL4. Composée de modules d'entrées/sorties et d'un concentrateur, elle s'appuie sur l'architecture bi-processeur pour assurer la gestion d'un réseau circulaire doublé de capteurs pouvant compter jusqu'à 128 entrées/sorties SIL4. Cette passerelle de communication est en cours de certification par CERTIFER.

Figure 6. Passerelle de communication SATURN SIL4 embarquant la technologie bi-processeur

La réduction de coût constatée, en plus du fait de rationaliser les développements en fournissant un socle commun à ceux-ci, est due au fait de n'avoir à développer qu'une seule instance du logiciel B (les codes binaires redondants et diversifiés étant obtenus par les deux chaines de génération de code HEX), et au fait que la phase de test se trouve réduite (grâce à la preuve mathématique de conformité spécification/conception). Enfin les fonctionnalités logicielles liées à la sécurité proprement dites ont été développées une seule fois et n'ont plus besoin d'être développées à nouveau.

La plateforme d'exécution a une puissance utilisable de 100 MIPS qui lui permet de réaliser des fonctions d'acquisition, traitement et commandes pour du contrôle commande portes palières, pour de la concentration de messages sur réseau sécurisé, etc. Les performances de la plateforme a donc été évaluées comme suffisantes pour des applications critiques embarquées et débarquées

8 Références

Lecomte, 2015, Architecture double cœur SIL4 pour automatismes sécuritaires à bas coût, Forum IFSTTAR XXI
Lecomte, 2016, LCHIP: Low Cost High Integrity Platform, Open Source Innovation Spring
Lecomte, 2016, Atelier B has turned 20, ABZ Conference