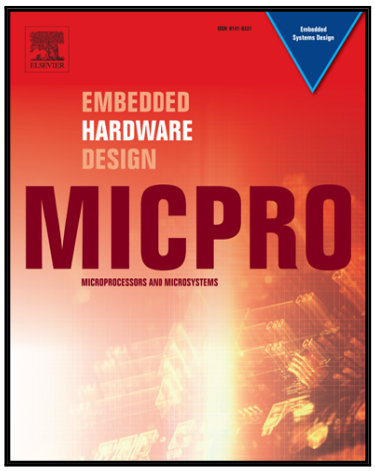

An End-to-end Framework for Safe Software Development

Mahmoud Hussein , Reda Nouacer , Ansgar Radermacher , Armand Puccetti , Christophe Gaston , Nicolas Rapin

PII: S0141-9331(18)30027-9
DOI: 10.1016/j.micpro.2018.07.004
Reference: MICPRO 2720

To appear in: *Microprocessors and Microsystems*

Received date: 22 January 2018
Revised date: 10 June 2018
Accepted date: 16 July 2018

Please cite this article as: Mahmoud Hussein , Reda Nouacer , Ansgar Radermacher , Armand Puccetti , Christophe Gaston , Nicolas Rapin , An End-to-end Framework for Safe Software Development, *Microprocessors and Microsystems* (2018), doi: 10.1016/j.micpro.2018.07.004

An End-to-end Framework for Safe Software Development

Mahmoud Hussein, Reda Nouacer, Ansgar Radermacher, Armand Puccetti,Christophe Gaston, Nicolas Rapin
CEA, LIST, Software and System Engineering Department (DILS),
P.C. 174, Gif-sur-Yvette, 91191, France
{mahmoud.hussein, reda.nouacer, ansgar.radermacher, armand.puccetti, christophe.gaston, and nicolas.rapin}@cea.fr

***Abstract*-It is largely recognized that the architectures of embedded systems are becoming more and more complex both at hardware and software levels. Despite the significant advances in the development tools, developing the software of such systems while ensuring their safety is still a difficult task. In this paper, we propose an end-to-end programming framework to ease the development of safe software systems. The programming framework, supported by a proper methodology and workflow, make it possible to design safe/secure software that implements functional requirements while respecting multiple non-functional requirements and mastering architectural complexity, time-to-market and cost. The programming framework is based on five concepts: (1) model-based system engineering: MBSE, (2) design-by-contract approach, (3) formal analysis of models based on symbolic execution, (4) code generation, and (5) static and dynamic code analysis. The effectiveness of the methodology has been demonstrated through multiple use-cases. The framework is realized using CEA LIST (http://www-list.cea.fr/en/) open-source development platforms: Papyrus, Frama-C, and UNISIM-VP. These platforms are results of many research and industrial projects such as FP7-SafeAdapt [1], FUI-EQUITAS [2], FP7-STANCE [3], CATRENE-OpenES [4], FSN-SESAM Grids [5], and H2020-VESSEDIA [6].**

Keywords- Embedded Systems; Model-driven Development; Safety Analysis; Simulation.

I. INTRODUCTION

It is largely recognized that the architectures of embedded systems are becoming more and more complex both at hardware and software levels. Thanks to the constant advances in microelectronic, embedded system engineers are now able to integrate more system functions on a powerful System-on-Chip (SoC) [1].Due to the high integration level, clock frequency, and functioning conditions (e.g. temperature, magnetic fields, etc.), the failure rate of a circuit increases by approximately√2in an eighteen month period[2]. In addition, the issue of robustness and reliability becomes crucial in the development of complex embedded software systems [3]. For instance, the respect of safety constraints is crucial for railway signaling, autonomous cars and robotic systems, to name just a few.

In the design of current critical embedded systems, two points of views are treated separately: the performance point of view and the safety/security point of view. Each of the two views uses models (for design and analysis) that are of a different nature. For systems such as avionics and automotive applications, it is crucial to have a "reliable performance". Therefore, the design must be integrated with a tool alignment or standardization (i.e. uniformization) of formalisms.

Therefore, there is a need for an environment that eases the development of such systems. It also should be kept integrated, and its tools (techniques) evolve hand in hand [4].An all in one programming framework supported by an industrial grade methodology is required to enable safe and efficient programming of multi-core and heterogeneous architectures. The programming framework has to support developers along the design workflow to produce efficient and safe/secure software system. The programming environment has to be a collaborative framework where different users' profiles may contribute and share their skills, when it is required, at different stages of the design workflow.

Recently, a number of research projects has been conducted to assist system engineers in developing complex embedded systems, while ensuring their safety(e.g. MultiPARTES [5], COMPLEX [6], IMPReSS[7], ACTORS [8], DREAMS [9], and SAFURE [10]). The projects have proposed development environments that are built on the partners' tools. But, keeping these environments fully working on the long term is difficult, where each partner's tool evolves separately.

CEA LIST has been a partner in a number of research and industrial projects, and has produced three complementary open-source development platforms: Papyrus [11], Frama-C [12], and UNISIM-VP [13][14]. In this paper, we propose an engineering methodology that uses these three platforms for developing safe software systems. The main objectives of our methodology are:

- Provide an end-to-end software development framework.
- Improve the efficiency of software developers.
- Increase the system efficiency, while ensuring its safety requirements.

[1] SafeAdapt project was funded by the European Commission within the 7th Framework Program under the grant number "608945".
[2] EQUITAS project was funded by Bpifrance under call FUI-AAP16 with a contract number "F1312031-Q".
[3] STANCE project was also funded by the European Commission under the ICT theme of the 7th Framework Program with the grant agreement number "317753".
[4] OpenES project was funded under the CATRENE Program with the agreement number "CA703–2013".
[5] SESAM Grids is a "Programme d'Investissement d'Avenir" project funded by "FSN-Briques Génériques du Logiciel Embarqué N°3" with contract number "O14830-67155".
[6] VESSEDIA project is funded by the European Commission under the SEC theme of the H2020 Framework Program with the grant agreement number "731453".