PhD thesis offer (CIFRE)

Mixing Unproved and Proved sub-systems through Contracts
for Correct-by-Construction system design

Company:	MITSUBISHI ELECTRIC R&D CENTRE EUROPE 1 allée de Beaulieu, CS 10806, 35708 Rennes Cedex 7, France Web site: http://www.mitsubishielectric-rce.eu/
Contract type:	3 years determined term period (CDD), September 2015
Reference:	FM_PhD_201510
Research theme:	Formal methods, systems and services

Scientific environment:

Mitsubishi Electric R&D Centre Europe (MERCE) is the European research laboratory of Mitsubishi Electric group. Mitsubishi Electric builds a wide range of products, from most common ones (fridges, air-conditioning, etc.) to most specialized safety critical ones (nuclear power plant or train control systems, satellites, power electronic systems, elevators, etc.). COM division of MERCE works on formal methods, amongst other topics, to improve product quality and reduce production costs while taking into account the whole development process (people qualification, properties to ensure, usability vs. provability ratio, integration into classical development process, etc.).

CEA LIST is a technological research center about software systems. It collaborates with major industrial companies in nuclear, automotive, aeronautics, defense and health areas. The Software Reliability and Security Laboratory (LSL) of CEA LIST, located at Saclay near Paris, aims to bridge the gap between academia and industries by implementing cutting-edge tools based on formal methods to validate and verify software in safety- and security-critical domains.

This PhD (3 years CIFRE contract) is proposed by the CEA LIST, located at Saclay near Paris.

Mitsubishi Electric R&D Centre Europe is going to jointly supervise the PhD.

Context of the PhD:

To meet commercial pressure, engineers need to develop systems in shorter time. On the contrary, to meet greater customer expectations or simply safety or regulatory constraints, the requirements on overall system quality are higher. Introduction of software has added a lot of flexibility to system design, allowing fulfilling partially the previous needs. However, industrial systems are more and more complex, being built from not only specific parts but also COTS (Components of the Shelf), mixing physical and software parts, etc. For example a factory automation system is made of pre-made

computing modules called PLC (Programmable Logic Controller), connected to actuators and detectors and interconnected to each other. Each PLC module should be programmed in such a way that the overall production chain makes a coherent action (e.g. peek a piece of equipment by a robot, put it on a machine, after machine processing, put back the piece of equipment on a trolley, etc.), while fulfilling global safety requirements. Moreover, to face rapid requirement changes and to be more efficient, more iterative system building are needed, where some properties of a system or sub-system are assumed but might be changed in future version of the architecture and design.

This increased complexity of system building makes the current approach more and more difficult. A possible solution would be to use a suitable method and tools that ensure Correct-by-Construction system architecture, design and implementation. In other words, the properties to fulfill are intrinsically weaved into the system during its construction. In this perspective, a promising way is the notion of "contracts". At any point of the system, e.g. module or component interface, one describes the expectations of both sides of the interface: the provider of the interface describes the provided service under needed requirements – pre-conditions – while the caller of the interface promise to fulfill those requirements with the guarantee to obtain the provided service – post-conditions. Such a contract can be both written by the end-user and generated by the development environment. A Meta-Theory of contracts has been defined[1] and several frameworks have been implemented like Frama-C for C language [2] or SPARK 2014 for Ada language [3].

Formal Methods also offer promising solutions to several of above issues by ensuring exhaustive verification of some properties. Amongst the wide variety of formal methods available, one approach is especially interesting because it allows verifying complex functional properties: the B Method [4]. It is based on the notion of *refinement*: starting from formal and non deterministic specifications, one progressively adds details down to reaching concrete code level that can be directly translated to machine code. Other more recent works on refinement are in progress (e.g. Leon Gondelman's PhD on refinement in Why3).

While being considered for a long time, the recent progresses on computer efficiency as well as on algorithmic aspects make the formal methods more and more practical to ensure Correct-by-Construction system building even if end-users are not formal methods specialists.

However, despite great increases in recent capabilities of formal methods, applying them is still a lot of work. Therefore there is a pressing need to only apply formal methods to only the minimal part of a system in order to ensure the most safety critical properties. This approach has been applied for 20 years now but mostly in an empirical way (e.g. Siemens subway control systems). Demonstrating that combining non-formal and formal parts builds a safe and secure system is done on paper, without any supporting tool. But, as built systems are more and more complex, this approach becomes less and less sustainable.

PhD Research proposal:

The main purpose of this PhD would be to build a framework where one can combine within the same system formally proved parts and non-formally proved ones, with the overall goal of formally proving global properties on the system. This goal is reachable, provided that the result of unproved parts is validated by formally proven checkers. This approach has already been applied and demonstrated viable in the past, e.g. register allocation in CompCert certified C compiler. Moreover, by using system architecture combining (complex) non-proved parts and (simplified) proved parts, adequate safety threshold can be reached in a demonstrable way (Littlewood & Rushby, 2012).

In order to reach this goal, three mechanisms would be used:

- A **Contract** formalism to specify the wanted properties from system level down to individual code part level;
- A **Refinement** system to provide abstraction and information hiding in order to apply this approach from individual hardware or software component up to complete system or system-of-systems;
- Management of **formally proven dynamic checkers** that would be at the interface between proven and unproven parts.

It is envisioned that proven and unproven parts could be intertwined, for example within the same code procedure. Moreover, in order to ensure obvious availability properties, a dynamic checker failing

verification should be taken in overall system design, i.e. by activating alternative (but still proven) paths within the system.

Therefore, the PhD student would have to define several points in order to reach this goal:

- Define a contract framework applicable from system to code level. This framework should take place within the Contract Meta-Theory of Benveniste et al., in order to ensure usability and scalability needed for system design. Moreover this contract framework should have an executable semantics, in order to easily mix proved and non-proved parts of a system;
- Define the mathematical conditions needed for the formal demonstration of a property at system level, provided that some elements of this property are ensured by formally proven parts while others are ensured by unproved parts augmented with formally proven dynamic checkers;
- Propose automatic synthesis of formally and automatically provable dynamic checkers for simple properties;
- Define a methodology and supporting tool for the effective application of such an approach in industrial system development.

Essential qualifications / personal profile:

- Recently graduated Engineer or University Master with a background in formal methods or Mathematics
- Fluent English
- Excellent communication skills
- Motivation and dynamism to work in a research environment
- Open-mindedness, capacity to work in a multicultural and international environment

Please send CV and motivation letter by email (indicating in object: your name + reference of the offer FM_PhD_201510) **to all following contacts**:

jobs@fr.merce.mee.com.

Julien.Signoles@cea.fr

virgile.prevosto@cea.fr

Bibliography:

Littlewood, B., & Rushby, J. (2012). Reasoning about the Reliability Of Diverse Two-Channel Systems In which One Channel is "Possibly Perfect". *IEEE Transactions on Software Engineering , 38* (5), 1178-1194.

[1] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger, and K. G. Larsen, “Contracts for system design,” 2012.

[2] F. Kirchner, N. Kosmatov, V. Prevosto, J. Signoles, and B. Yakobowski. Frama-C: A software Analysis Perspective. Formal Aspects of Computing, pages 1-37, January 2015.

[3] Spark2014. http://www.spark-2014.org/

[4] J.-R. Abrial. The B-book: assigning programs to meanings. Cambridge University Press. 1996.