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

M2 internship subject proposal - Version 1 – 2015-11-02

Context

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, put back the piece of equipment on a trolley after machine processing, 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.

M2 internship research proposal

The main purpose of this M2 internship would be to start building 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 [5].

An important capability of the demonstration is to take into account degraded modes, where in certain modes only a subset of the overall properties are formally verified, through for example de-activation of some modules or activation of by-passes.

During its M2 internship, the student will have to tackle the following points:

- As a first but representative use case, develop using the C language the Landing Gear System [6]. Prove some properties on it using the Frama-C framework;
- Through literature and industrial case studies, elaborate a list of composition schemes used to mix formal and non-formal parts in order to gain formal properties in a system;
- Elaborate in Contract Meta-Theory of Benveniste et al. [1] a formal framework that allows to describe and to verify formal properties on a mix of formal and non formal modules. This framework should take into account the use of degraded modes, such that only certain formal properties are verified in certain modes;
- (If time permit) define and make a prototype tool that implements the defined framework;
- Apply the defined framework on the Landing Gear System example.

This M2 internship could be continued into a PhD if results are satisfying enough.

Scientific environment

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.

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.).

Applying

Candidate must be enrolled in a Master 2 curriculum (or equivalent) in Computer Science, with emphasis in programming languages, formal methods, and/or software engineering. Experience in C programming and/or model-driven development would be a plus. To apply, send a resume and a cover letter to Virgile Prevosto (virgile.prevosto@cea.fr), Julien Signoles (julien.signoles@cea.fr), Benoît Boyer (B.Boyer@fr.merce.mee.com) and David Mentré (d.mentre@fr.merce.mee.com).

The internship is paid. Work will be carried out at CEA LIST's Nano-Innov site in Palaiseau (20km South of Paris). Housing stipend is possible under conditions. Intern will have access to CEA's shuttle service for commuting.

Please note that CEA can take up to 3 months for instructing a file, thus be sure to apply as early as possible.

Bibliography

[1] A. Benveniste, B. Caillaud, D. Nickovic, R. Passeroneau, 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,* pp. 1-37, 2015.

[3] "Spark2014," [Online]. Available: http://www.spark-2014.org/.

[4] J.-R. Abrial, The B-Book: Assigning Programs to Meanings, Cambridge University Press, 1996.

[5] B. Littlewood and J. Rushby, "Reasoning about the Reliability Of Diverse Two-Channel Systems In which One Channel is "Possibly Perfect"," *IEEE Transactions on Software Engineering,* vol. 38, no. 5, pp. 1178-1194, 2012.

[6] V. Wiels and F. Boniol, "Landing Gear System, Case study for the ABZ 2014 conference," in *ABZ conference*, 2014.