2018-00549 - Formal aspects of multiview, dynamic hardware and software architectures [PhD Campaign]

Level of qualifications required: Graduate degree or equivalent
Fonction: PhD Position

About the research centre or Inria department
Grenoble Rhône-Alpes Research Center groups together a few less than 800 people in 35 research teams and 9 research support departments.

Staff is localized on 5 campuses in Grenoble and Lyon, in close collaboration with labs, research and higher education institutions in Grenoble and Lyon, but also with the economic players in these areas.

Present in the fields of software, high-performance computing, Internet of things, image and data, but also simulation in oceanography and biology, it participates at the best level of international scientific achievements and collaborations in both Europe and the rest of the world.

Context
Modern embedded system architectures, such as those developed using the AUTOSAR standard, involve multiple views of a given system, including logical views of application components, and physical views encompassing the assembly and mapping of application components on a given hardware architecture. In addition, such architectures combine functional specifications with quantitative constraints, such as real-time bounds for task execution and component reliability requirements.

Despite a rich literature on the subject (see [1] for a recent survey), multiview modeling in hardware/software design and co-design is still poorly understood, and its support in system modeling or architectural description languages is still largely unsatisfactory [2].

The Spades project-team at INRIA Grenoble-Rhône-Alpes is currently developing a process calculus framework, called Location Graphs [3], that aims to provide a core formal basis for modeling modular and dynamic architectures. In contrast to other formal component-based approaches, location graphs support arbitrary graphs as component configurations, different composition operators for each graph node, and direct support for different coexisting views for the same system configuration. From a more practical point of view, the location graph framework can be seen as an evolution and extension of the Fractal component framework [4].

Assignment
The main objectives of the PhD will be twofold:
1. To further the development of the location graph approach to the multiview modeling of dynamic hardware and software architectures, and of its behavioral theory.
2. To develop static analyses for location graphs addressing both behavioral and structural aspects.
We expect the work to encompass both theoretical investigations as well as prototype tool developments.
Main activities

The theoretical developments may involve the following:
- Develop the location graph behavioral theory (notions of equivalence, simulation and refinement), for which recent developments on contextual equivalence for higher-order calculi can be leveraged [5,6,7].
- Relate the location graph model with other frameworks in different relevant domains, such as SHR systems for dynamic service modeling [8], Asynchronous Relational Nets [9], or Guarded Atomic Actions for hardware/software co-design [10].
- Develop a type system for location graphs, encompassing behavioral and architectural constraints (such as co-location, encapsulation and partitioning of components), drawing insights e.g. from behavioral types [11], semantic subtyping for graphs [12], or types for component assemblies [13].
- Develop adequate logics and proof techniques for compositional and multiview reasoning with location graphs, drawing insights e.g. from spatial logics and separation logics for concurrency [14,15,16].

We expect these theoretical developments to be machine-checked using the Coq proof assistant.

The tool developments may involve the following:
- The development of type checking and consistency checking analyses for location graphs.
- The development a prototype architecture description language exploiting typed location graphs at its core.

REFERENCES


Skills

- A good background on concurrency theory, programming language semantics and type systems is required.
- For the development work, a good knowledge of a modern programming language such as OCaml is required.
- A working knowledge of the Coq proof assistant would be appreciated.

Benefits package

- Subsidised catering service
- Partially-reimbursed public transport
- Social security
- Paid leave
- Flexible working hours
- Sports facilities