Postdoctoral Position

Scalable Model Checking of Embedded Real-Time Systems

The Real-Time Systems Group at the *Max Planck Institute for Software Systems* (MPI-SWS) seeks a motivated postdoctoral researcher to work on the subject of scalable model checking of embedded real-time systems. The research will be done in collaboration with the Modeling and Verification team at the *Institut de Recherche en Informatique Fondamentale IRIF* (CNRS and Université Paris Cité, France).

1 Context and Motivation

Embedded Real-Time Systems (ERTS) are at the heart of many safety-critical applications in diverse domains such as transportation, robotics, and healthcare, where their failure can cause significant economic damage and even loss of life. Formal verification of ERTSs to ensure their safe behavior is therefore crucial. Such verification is typically performed on an ERTS model with respect to important *real-time guarantees*, such as bounded response times or maximum data-age constraints. Both the real-time systems and formal methods communities have long studied related problems and developed elegant verification techniques, but all known techniques still face significant challenges in practice, and none is a panacea. For example, model checking (formal methods) is often limited by the state-space explosion problem, while schedulability analysis (real-time systems) often works with relatively coarse-grained, special-purpose models and thus can be inherently pessimistic and limited in generality.

The *schedule abstraction graph* (SAG) method [6, 7, 9] is a family of state-of-the-art schedulability analyses that apply well-known model-checking techniques, in particular on-the-fly reachability analysis, specifically tailored to the problem of providing per-job response-time bounds. The benefit of SAG's problem-specific specialization of the underlying model-checking techniques is improved *scalability*: in previous evaluations using the best-known models of real-time execution found in the literature *at the time*, SAG was found to be significantly faster and to scale to much larger problem sizes than the state-of-the-art model checker UPPAAL [5], sometimes by several orders of magnitude.

However, SAG's major drawback is a complete loss of generality: real-time properties of interest other than per-job response times, such as bounds on the temporal separation of events occurring on different processors, cannot be accurately inferred with SAG. In addition, a recent approach [3] relying on *new, much more carefully designed UPPAAL models* of real-time execution calls into question the scalability advantages, if any, of SAG over model-checking approaches that retain full generality.

It is therefore of great interest to revisit the question of scalable model checking of embedded real-time systems: is there much benefit to problem-specific tailoring of model checking algorithms in the manner of SAG, or is it possible to achieve similar (or even better) scalability *without loss of generality* by careful selection of models?

Accordingly, this postdoctoral position revolves around three major tasks:

- Perform a rigorous, up-to-date, and in-depth comparison between SAG and general model-checking approaches. The research will be based on two main observations. First, since SAG is based on *discrete time*, the candidate will first investigate the use of prominent *untimed* model checkers as a baseline. Particular attention will be paid to the LTSmin toolset [4], which provides state-of-the-art optimizations for partial-order reduction and parallel model checking. Other candidates such as NuXmv [2], ITS-Tools [8] and TINA [1] will also be considered. Second, since scalability depends heavily on the modeling phase, special attention will be given to model optimization (besides correctness). UPPAAL models, for instance from [3], will be used as a basis.

- Based on the results of the previous task, develop specific model checking algorithms for different classes of ERTSs, e.g., depending on the task model, the data sharing scheme, and the scheduling policy of the system. These algorithms, which may be partly inspired by SAG, will be integrated into a suitable model checker, such as LTSmin. It is envisioned that the genericity of the model checker can be preserved, while problem-specific

reduction techniques can greatly enhance the scalability of the model checker in the ERTS domain. A model checking toolbox with several verification options will be implemented accordingly.

- Finally, the postdoctoral researcher will evaluate the toolbox developed in the previous task on real-world case studies, such as those given in [3]. Based on the empirical observations, they will further derive guidelines on the most scalable technique(s) to use depending on the class of ERTS at hand.

2 Candidate, Duration and Location

The postdoctoral researcher will be primarily based at the Max Planck Institute for Software Systems (MPI-SWS) at the institute's branch in Kaiserslautern, Germany. It is expected that the collaborative nature of the project will require frequent short visits to IRIF (Paris, France).

The initial funding period is one year, with the possibility of an extension to two years. It is desired that the position be filled as soon as possible, but other arrangements are possible.

The candidate should have completed (or be nearing completion of) a Ph.D. in computer science or a closely related field. Motivation, curiosity, and basic knowledge of automata theory and/or real-time scheduling are required; experience with model checking is highly desirable. Programming skills are required for the planned empirical evaluation; experience with Linux shell scripting to automate and orchestrate large-scale experiments is a plus.

3 Contact

For any questions regarding this position and to apply, please email both:

- Dr. Björn Brandenburg (MPI-SWS) (bbb@mpi-sws.org)
- Dr. Mohammed Foughali (IRIF, Université Paris Cité) (foughali@irif.fr)

When applying for the position, please include your **full academic CV**, your (up to) three most significant or most project-relevant **publications**, and some **work samples** demonstrating your applied skills (e.g., major software development projects, open-source contributions, artifact evaluation submissions, etc.). Please attach either copies of your publications and ZIP archives of your work samples or provide links to public versions on the Web.

References

[1] Bernard Berthomieu, P-O Ribet, and François Vernadat. The tool TINA–construction of abstract state spaces for Petri nets and time Petri nets. *International journal of production research*, 42(14):2741–2756, 2004.

[2] Roberto Cavada, Alessandro Cimatti, Michele Dorigatti, Alberto Griggio, Alessandro Mariotti, Andrea Micheli, Sergio Mover, Marco Roveri, and Stefano Tonetta. The nuXmv symbolic model checker. In *International Conference on Computer Aided Verification (CAV)*, pages 334–342. Springer, 2014.

[3] Mohammed Foughali, Pierre-Emmanuel Hladik, and Alexander Zuepke. Compositional verification of embedded real-time systems. *Journal of Systems Architecture. PDF*, 2023.

[4] Gijs Kant, Alfons Laarman, Jeroen Meijer, Jaco Van de Pol, Stefan Blom, and Tom Van Dijk. LTSmin: high-performance language-independent model checking. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 692–707. Springer, 2015.

[5] Kim G. Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a nutshell. *Int. Jour. on Software Tools for Technology Transfer (STTT)*, 1(1):134–152, 1997.

[6] Mitra Nasri and Björn B. Brandenburg. An exact and sustainable analysis of non-preemptive scheduling. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 12–23, 2017.

[7] Mitra Nasri, Geoffrey Nelissen, and Björn B. Brandenburg. Response-time analysis of limited-preemptive parallel dag tasks under global scheduling. In *Euromicro Conference on Real-Time Systems (ECRTS)*, pages 21–1, 2019.

[8] Yann Thierry-Mieg. Symbolic model-checking using ITS-tools. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 231–237. Springer, 2015.

[9] Beyazit Yalcinkaya, Mitra Nasri, and Björn B. Brandenburg. An exact schedulability test for non-preemptive self-suspending real-time tasks. In *Design, Automation Test in Europe (DATE)*, pages 1228–1233, 2019.