Evaluation and Comparison of Real-Time Systems Analysis Methods and Tools

Sophie Quinton

Evaluation and Comparison of Real-Time Systems Analysis Methods and Tools*

Sophie Quinton

Univ. Grenoble Alpes, INRIA, CNRS, Grenoble INP, LIG, Grenoble, France

Abstract. The verification of real-time systems has been an active area of research for several decades now. Some results have been successfully transferred to industry. Still, many obstacles remain that hinder a smooth integration of academic research and industrial application. In this extended abstract, we discuss some of these obstacles and ongoing research and community efforts to bridge this gap. In particular, we present several experimental and theoretical methods to evaluate and compare real-time systems analysis methods and tools.

Keywords: Real-time systems · Verification · Formal methods.

1 Introduction

Critical embedded systems such as cars, satellites or planes are real-time in the sense that they must provide some type of timing guarantees, e.g., to ensure that a system will always react sufficently quickly to some external event.

The verification of real-time systems has been an active area of research for several decades now since the seminal work of Liu and Layland [20] (see [8] for a survey). Some results have been successfully transferred to industry, as illustrated by the existence of numerous companies selling real-time systems analysis tools which are spin-offs from research institutions, e.g., AbsInt[1] (from Saarland University), Symtavision[2] (now part of Luxoft, from TU Braunschweig) and RTaW[3] (from INRIA). Four additional examples of successful technology transfer are described in [7].

Still, many obstacles remain that hinder a smooth integration of academic research and industrial application. To illustrate this on an example, the verification of timing properties in the automotive industry tends to be based on simulations rather than static analysis, complemented with monitoring to handle at runtime potential timing violations. The rapid evolution of real-time systems, with the advent of multicore architectures and the shift toward heterogeneous,

* This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

[1] https://www.absint.com/

[2] https://auto.luxoft.com/uth/timing-analysis-tools/

[3] http://www.realtimeatwork.com

high-performance platforms, is increasing the gap between the analysis tools and methods proposed by the research community and the needs of industry [25].

At the same time, this trend represents a unique opportunity, because systems are becoming so complex that simulation is not a viable verification method anymore. There is currently a need for simple mechanisms to make these new, complex platforms more predictable, and for associated verification techniques. For example, several automotive OEMs and suppliers are now using the Logical Execution Time (LET) [18] paradigm to achieve predictable communication [17,14]. This choice has led to renewed interactions between academia and industry in order to identify where more research is needed on the topic [9].

In this context, we argue that one major obstacle to the application of academic results in industry is the difficulty, both for academics and practitioners, to evaluate how existing analysis techniques and their associated tools can perform on real systems. In the following, we discuss some criteria for such an evaluation that deserve more attention from the research community. We then present current efforts toward experimental and theoretical methods to evaluate and compare real-time systems analysis methods and tools.

2 Evaluation criteria

In this section, we would like to draw attention to several criteria that are key to evaluating the usability of a method or a tool, and which we feel are currently underestimated.

2.1 Expressivity of the underlying model

One major difficulty that practitioners face whenever trying to use a tool from academia, e.g. pyCPA[4], MAST[5] or Cheddar [6], is the mismatch between the models they work with and the expressivity of the tool they would like to use [16]. Many papers still assume a simple model where independent software tasks execute on a uniprocessor. In practice, systems are now much more complex, with multiple cores, communication buses, shared caches, etc. Even uniprocessor systems require more complex models than the one introduced in [20].

One example is illustrated in [15]: Due to minor uncertainties in clock implementations, the exact value of a task period (describing the frequency with which the task is activated) may not be known. This means that the activations of two tasks that are specified with the same period may shift if mapped onto different processors, which must be taken into account by the analysis. This requires support for parameters in the system model.

Another example related to the description of task activations concerns tasks implementing engine control. Such tasks, which are commonly referred to as adaptive variable-rate (AVR) tasks, are activated whenever the crankshaft of the

[4] https://pycpa.readthedocs.io
[5] https://mast.unican.es/mast.html
[6] http://beru.univ-brest.fr/~singhoff/cheddar/

engine reaches a specific angle. Recent theoretical works [22,3] provide solutions for precisely analyzing such tasks, but most tools do not implement them.

These are just two examples to illustrate the complexity of modeling industrial systems. Reality is even more complex, with tasks often implemented with some degree of intra-task parallelism [23], and the additional complexity due to multicore architectures. Many existing tools and analysis techniques do take into account some level of complexity in their model, including those cited above, but they apply to different, incomparable models with no systematic way of comparing them. Despite existing efforts [21], we still lack a clear understanding of how different abstractions can be compared semantically.

2.2 Expressivity of the provided guarantees

A second, related issue is the fact that academic research has largely focused on guaranteeing schedulability, that is, on ensuring that no task in a given task set can ever miss its deadline. Schedulability is usually established by computing an upper bound on the worst-case response time of tasks, i.e., the maximal delay between the activation of a task resulting in the creation of a job to be executed, and the completion of that job. This is often not the most critical issue.

First, one is generally not interested in the response time of a single task, but rather in the end-to-end latency of a so-called cause-effect chain of tasks which are independently activated but communicate via shared variables [10]. Although this problem was formalized ten years ago, it has only recently become an active research topic [1].

Besides, the notion of schedulability itself (even if the notion of deadline is applied to cause-effect chains rather than single tasks) is restrictive [2]: In particular, it has been shown that many real-time systems are weakly-hard rather than hard, meaning that they can tolerate a bounded number of deadline misses without this leading to system failure [11,19].

These two examples illustrate the fact that researchers and tool providers must pay closer attention to which timing guarantees are used in practice. A better understanding about how the real-time aspects interface with other viewpoints such as function or energy consumption is needed for that [13].

2.3 Precision of the computed results

Another problem that hinders the use of academic solutions for the verification of real-time systems is the lack of support to estimate the precision of the computed results. Indeed, for scalability reasons, existing solutions compute upper bounds on worst-case behaviors, which introduces some pessimism in the analysis. The problem is that there is no method to quantify that pessimism, other than comparing the computed upper bounds with results obtained by simulation. In general, there is a large gap between the values thus obtained (through analysis and simulation) and the user does not know whether it is due to the imprecision of the simulation, or whether it results from the pessimism of the

analysis. Exhibiting a possible scenario leading to a deadline miss would be valuable to practitioners because it would help them to redesign the system to make it schedulable. The need to investigate this issue further and some initial results are provided in [12].

3 Methods for evaluation and comparison

Let us now present several initiatives aiming to help researchers and practitioners compare their methods and tools.

3.1 Empirical approaches

The objective of the WATERS industrial challenge[7] is to address the need for closer interaction between academia and industry that is underlined by the observations made in the previous sections. The principle of the challenge is to provide researchers with a concrete industrial problem related to real-time systems design and analysis, which they try to solve with their preferred method and tool. So far, Thales, Bosch and Dassault have contributed (Bosch has proposed multiple challenges). The WATERS industrial challenge has proven over the years to be an extremely attractive and valuable exercice to share and compare solutions and results.

While we need more case studies such as the WATERS industrial challenge, we also need synthetic test cases, or tools to generate them, on which there is a consensus. Unfortunately, there is no such tool at the moment – authors use custom made generators for their publications. Some rules to generate meaningful test cases are provided in [6], but the targeted models are too simple to tackle realistic sytems and need to be extended.

RTSpec [24] represents a significant effort towards a unified format for describing such test cases. It is a formalism for real-time systems specification with flexible syntax and rigorous semantics based on UPPAAL models. Based on this library, the timing model of various analyzers can be formalized, and mappings between their respective input formats can be rigorously defined. The overall target is a framework which comprises the RTSpec formalism, a tool chain for automatically translating RTSpec into the input of various analysis tools, and a set of benchmarks which are synthetic or derived from industrial case studies. Such a framework would provide a systematic, automated and rigorous methodology for evaluating analyzers.

3.2 Theoretical approaches

Few research papers have focused on the issue of comparing real-time systems analysis techniques. A recent publication [5] (building upon [6]) is tackling the problem while identifying pitfalls in the use of metrics such as resource augmentation factors and utilization bounds to compare methods or tools.

[7] https://www.ecrts.org/industrialchallenge

One theoretical tool which seems promising to provide a solid formal background for comparing models and analysis techniques is the Prosa[8] library, a repository of definitions and proofs for real-time schedulability analysis [4] using the Coq proof assistant[9]. One of the objectives of the ongoing CASERM project[10] is to build the RTSpec framework on top of Prosa instead of UPPAAL, thus allowing for formal proofs on model transformations, as needed for comparison purposes.

4 Conclusion

In this short paper, we have illustrated the need for a better theoretical and practical support to evaluate and compare methods and tools for real-time systems analysis. We have underlined the importance of being able to formally relate models used by different approaches, as well as the need to look beyond schedulability analysis and to develop methods to quantify the pessimism of existing analyses. In addition, we have presented recent and ongoing initiatives targeting these goals, which we hope will help reducing the gap between academic research and industrial practice.

References

1. Becker, M., Dasari, D., Mubeen, S., Behnam, M., Nolte, T.: End-to-end timing analysis of cause-effect chains in automotive embedded systems. Journal of Systems Architecture - Embedded Systems Design **80**, 104–113 (2017), https://doi.org/10.1016/j.sysarc.2017.09.004
2. Beyond the deadline: New interfaces between control and scheduling for the design and analysis of critical embedded systems. Tutorial at ESWeek 2017: https://team.inria.fr/spades/beyond-the-deadline/ (2017)
3. Biondi, A., Natale, M.D., Buttazzo, G.C.: Response-time analysis of engine control applications under fixed-priority scheduling. IEEE Trans. Computers **67**(5), 687–703 (2018), https://doi.org/10.1109/TC.2017.2777826
4. Cerqueira, F., Stutz, F., Brandenburg, B.B.: PROSA: A case for readable mechanized schedulability analysis. In: 28th Euromicro Conference on Real-Time Systems, ECRTS 2016. pp. 273–284 (2016), https://doi.org/10.1109/ECRTS.2016.28
5. Chen, J.J., von der Brüggen, G., Huang, W.H., Davis, R.I.: On the pitfalls of resource augmentation factors and utilization bounds in real-time scheduling. In: 29th Euromicro Conference on Real-Time Systems, ECRTS 2017. vol. 76, pp. 9:1–9:25 (2017), https://doi.org/10.4230/LIPIcs.ECRTS.2017.9
6. Davis, R.I.: On the evaluation of schedulability tests for real-time scheduling algorithms. In: 7th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems, WATERS 2017 (2017)

[8] http://prosa.mpi-sws.org
[9] https://coq.inria.fr/
[10] https://project.inria.fr/caserm

7. Davis, R.I., Bate, I., Bernat, G., Broster, I., Burns, A., Colin, A., Hutchesson, S., Tracey, N.: Transferring real-time systems research into industrial practice: Four impact case studies. In: 30th Euromicro Conference on Real-Time Systems, ECRTS 2018. pp. 7:1–7:24 (2018), https://doi.org/10.4230/LIPIcs.ECRTS.2018.7
8. Davis, R.I., Burns, A.: A survey of hard real-time scheduling for multiprocessor systems. ACM Computing Surveys (CSUR) **43**(4), 35 (2011)
9. Ernst, R., Kuntz, S., Quinton, S., Simons, M.: The logical execution time paradigm: New perspectives for multicore systems (dagstuhl seminar 18092). Dagstuhl Reports **8**(2), 122–149 (2018), https://doi.org/10.4230/DagRep.8.2.122
10. Feiertag, N., Richter, K., Nordlander, J., Jonsson, J.: A compositional framework for end-to-end path delay calculation of automotive systems under different path semantics. In: 1st Workshop on Compositional Theory and Technology for Real-Time Embedded Systems, CRTS 2008 (2008)
11. Frehse, G., Hamann, A., Quinton, S., Woehrle, M.: Formal analysis of timing effects on closed-loop properties of control software. In: Proceedings of the IEEE 35th IEEE Real-Time Systems Symposium, RTSS 2014. pp. 53–62 (2014), http://dx.doi.org/10.1109/RTSS.2014.28
12. Girault, A., Henia, R., Prévot, C., Quinton, S., Sordon, N.: Improving and estimating the precision of bounds on the worst-case latency of task chains. In: ACM SIGBED International Conference on Embedded Software, EMSOFT 2018 (2018), To appear.
13. Graf, S., Quinton, S., Girault, A., Gössler, G.: Building correct cyber-physical systems: Why we need a multiview contract theory. In: 23rd International Conference on Formal Methods for Industrial Critical Systems, FMICS 2018 (2018), To appear.
14. Hamann, A., Dasari, D., Kramer, S., Pressler, M., Wurst, F.: Communication centric design in complex automotive embedded systems. In: 29th Euromicro Conference on Real-Time Systems, ECRTS 2017. pp. 10:1–10:20 (2017), https://doi.org/10.4230/LIPIcs.ECRTS.2017.10
15. Henia, R., Rioux, L.: WATERS industrial challenge by Thales. https://www.ecrts.org/industrialchallenge-thales
16. Henia, R., Rioux, L., Sordon, N., Garcia, G., Panunzio, M.: Integrating model-based formal timing analysis in the industrial development process of satellite on-board software. In: 2nd International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2014. pp. 619–625 (2014), https://doi.org/10.5220/0004874306190625
17. Hennig, J., von Hasseln, H., Mohammad, H., Resmerita, S., Lukesch, S., Naderlinger, A.: Towards parallelizing legacy embedded control software using the LET programming paradigm. In: WiP session at the 2016 IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2016. p. 51 (2016), http://dx.doi.org/10.1109/RTAS.2016.7461355
18. Kirsch, C.M., Sokolova, A.: The logical execution time paradigm. In: Advances in Real-Time Systems. pp. 103–120 (2012), http://dx.doi.org/10.1007/978-3-642-24349-3_5
19. Linsenmayer, S., Allgöwer, F.: Stabilization of networked control systems with weakly hard real-time dropout description. In: 56th IEEE Annual Conference on Decision and Control, CDC 2017. pp. 4765–4770 (2017). https://doi.org/10.1109/CDC.2017.8264364, https://doi.org/10.1109/CDC.2017.8264364
20. Liu, C.L., Layland, J.W.: Scheduling algorithms for multiprogramming in a hard-real-time environment. J. ACM **20**(1), 46–61 (1973), http://doi.acm.org/10.1145/321738.321743

21. Long, A.B., Ouhammou, Y., Grolleau, E., Fejoz, L., Rioux, L.: Bridging the gap between practical cases and temporal performance analysis: a models repository-based approach. In: Proceedings of the 25th International Conference on Real-Time Networks and Systems, RTNS 2017. pp. 178–187 (2017), http://doi.acm.org/10.1145/3139258.3139286
22. Mohaqeqi, M., Abdullah, J., Ekberg, P., Yi, W.: Refinement of Workload Models for Engine Controllers by State Space Partitioning. In: 29th Euromicro Conference on Real-Time Systems, ECRTS 2017. vol. 76, pp. 11:1–11:22 (2017), http://10.4230/LIPIcs.ECRTS.2017.11
23. Serrano, M.A., Melani, A., Kehr, S., Bertogna, M., Quiñones, E.: An analysis of lazy and eager limited preemption approaches under dag-based global fixed priority scheduling. In: 20th IEEE International Symposium on Real-Time Distributed Computing, ISORC 2017. pp. 193–202 (2017), https://doi.org/10.1109/ISORC.2017.9
24. Shan, L., Graf, S., Quinton, S., Fejoz, L.: A framework for evaluating schedulability analysis tools. In: Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday. pp. 539–559 (2017). https://doi.org/10.1007/978-3-319-63121-9_27, https://doi.org/10.1007/978-3-319-63121-9_27
25. Simon Kramer, Dirk Ziegenbein, A.H.: Real world automotive benchmark for free. In: 5th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems, WATERS 2015 (2015)