Real-time Analysis of Resource-Constrained Distributed Systems by Simulation-Guided Model Checking

Gabor Madl Nikil Dutt
Center for Embedded Computer Systems
University of California, Irvine, CA 92697, USA
{gabe, dutt}@ics.uci.edu

Abstract

This research proposes a model-based design flow for the dynamic real-time analysis of resource-constrained distributed systems. We propose an analysis framework that combines simulations and model checking for performance evaluation and real-time verification. Our results show that the proposed method can provide better coverage for dynamic analysis than alternative methods, and can improve the accuracy of static analysis methods for performance estimation.

1 Introduction

As real-time systems are becoming increasingly distributed and embedded, designers need to address a growing number of QoS constraints that have traditionally been relevant to hardware design, such as real-time properties, end-to-end performance and energy consumption.

Classical scheduling theory provides scalable methods for real-time analysis, but is often too pessimistic and inaccurate for performance estimation. This research argues that to address challenges in resource-constrained distributed real-time embedded (DRE) systems, designers need to combine static analysis methods with multiple dynamic analysis methods, such as simulations and model checking, to improve the accuracy of the analysis.

We propose a real-time analysis method for distributed real-time systems by simulation-guided model checking [8]. We propose a design flow using the concept of model-integrated computing [12] to capture simulations and model checking in a formal framework, and demonstrate the concept on the domain of DRE systems in Section 3. Our results show that the combination of simulations and model checking improves the existing practice of static analysis and ad-hoc simulations for the early estimation of real-time properties in component-based DRE systems.

2 Related Work

Classic scheduling theory provides real-time analysis methods for distributed systems [4, 11]. Synchronous languages [2] are an alternative approach to specify timing constraints. Some methods exist that extend scheduling theory [10] and real-time calculus [13] for performance estimation. The disadvantage of static analysis methods is that they are often overly pessimistic, and therefore less accurate than dynamic analysis methods. Moreover, they cannot provide counter-examples when real-time properties are violated. The current industry practice for the dynamic performance evaluation of large-scale real-time systems is mostly based on simulations. Although simulations can capture single execution traces of a system accurately, they do not cover all execution traces and therefore cannot guarantee real-time properties in general. Model checking is an alternative approach for the dynamic analysis of distributed real-time systems. Task timed automata models were proposed in [3] to verify real-time scheduling. Model checking methods, however, often suffer from the state space explosion problem and limit exploration to answering yes/no questions. Therefore, we need a combination of methods to achieve the best coverage for performance estimation.

3 Model-based Analysis of Resource-constrained DRE Systems

We propose a model-based analysis method for the verification and dynamic performance estimation of resource-constrained distributed real-time embedded (DRE) systems using the concept of model-integrated computing [12], as shown in Figure 1. The design flow starts with the domain-specific model (DSM), a high-level specification that captures key properties of the design, such as its structure, behavior, environment, and key constraints that it has to satisfy. The domain-specific model can be expressed in several ways, using architecture description languages (ADLs),

timing diagrams, meta-modeling, or other visual methods. The DSM is then mapped to a formal semantic domain, that provides an executable formalism for the analysis. By utilizing an abstract formal representation of the system for simulations, significant simulation speedups can be achieved for dynamic analysis. As execution parameters are obtained by simulations, the accuracy of the formal analysis is comparable to simulation results, while providing better coverage. The formal executable model can be mapped to heterogeneous models of computation (MoC). This approach allows to find the formalism that provides the best scalability with the required precision. Moreover, unlike pure model checking methods, the proposed approach can provide partial simulation results in cases where exhaustive analysis is infeasible. Therefore, the combination of simulations and model checking improves the existing practice of random simulations and can provide partial results when model checking methods fail due to the state space explosion problem.

We have applied an early version of the model-based analysis framework shown in Figure 1 for the verification of real-time CORBA DRE avionics applications using timed automata in [6]. We generalized this concept to medium-size distributed systems by introducing FIFO buffers and fixed-priority schedulers in [7]. We described an approach for the real-time verification of preemptive scheduling in [5]. We introduced the concept of combining simulations and finite state machine model checking for the performance estimation of Multi-Processor System-on-Chip (MPSoC) designs in [9], and have uncovered an ambiguity in the AMBA AHB specification [1]. Recently, we proposed a method for simulation-guided real-time verification and performance estimation of DRE systems using discrete event simulations in [8]. The proposed design flow shown in Figure 1 was implemented in the open-source DREAM tool, that was recently integrated in the *Scenery* project by *Fujitsu Laboratories of America*, for the real-time analysis of deeply embedded systems, and is available at `http://dre.sourceforge.net`.

4 Conclusion

The proposed dynamic analysis framework shown in Figure 1 provides a design flow for the real-time analysis of resource-constrained DRE systems by combining simulations and model checking methods. Advantages of the approach include *(1)* improved accuracy compared to static performance estimation methods, *(2)* increased coverage compared to dynamic simulation-based methods, *(3)* the ability to provide partial results when model checking fails due to the state explosion problem, and *(4)* the capability to provide counter-examples when real-time properties are violated.

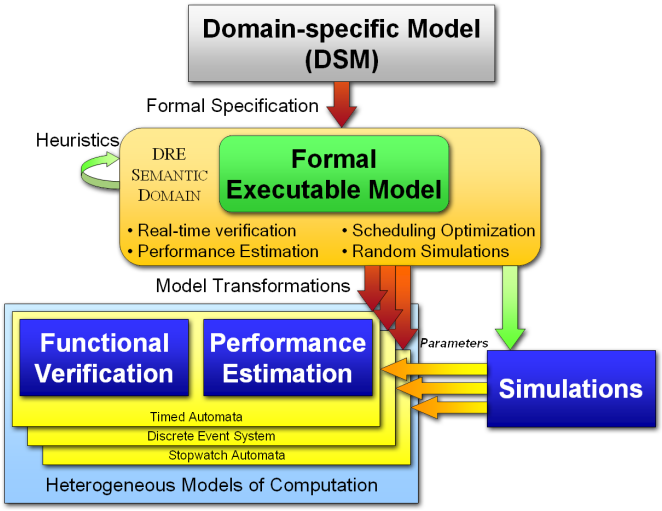

Figure 1. Model-based Analysis Framework

References

[1] ARM. AMBA Specification rev 2.0, IHI-0011A, 1999.

[2] A. Benveniste et al. The Synchronous Languages 12 Years Later. *Proceedings of the IEEE*, 91:64–83, 2003.

[3] C. Ericsson et al. Timed Automata as Task Models for Event-Driven Systems. In *Proceedings of RTSCA*, pages 182–189, 1999.

[4] C. L. Liu and J. W. Layland. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *J. ACM*, 20(1):46–61, 1973.

[5] G. Madl and S. Abdelwahed. Model-based Analysis of Distributed Real-time Embedded System Composition. In *Proceedings of EMSOFT*, 2005.

[6] G. Madl et al. Automatic Verification of Component-Based Real-Time CORBA Applications. In *Proceedings of RTSS*, pages 231–240, December 2004.

[7] G. Madl et al. Verifying Distributed Real-time Properties of Embedded Systems via Graph Transformations and Model Checking. *Real-Time Systems*, 33:77–100, Jul 2006.

[8] G. Madl et al. Performance Estimation of Distributed Real-time Embedded Systems by Discrete Event Simulations. In *Proceedings of EMSOFT*, October 2007.

[9] G. Madl et al. Formal Performance Evaluation of AMBA-based System-on-Chip Designs. In *Proceedings of EMSOFT*, pages 311–320, October 2006.

[10] K. Richter et al. A Formal Approach to MpSoC Performance Verification. *IEEE Computer*, 36:60–67, April 2003.

[11] L. Sha et al. Real Time Scheduling Theory: A Historical Perspective. *Real-Time Systems*, 28:101–155, November–December 2004.

[12] J. Sztipanovits and G. Karsai. Model-Integrated Computing. *IEEE Computer*, pages 110–112, Apr. 1997.

[13] E. Wandeler et al. System architecture evaluation using modular performance analysis - a case study. *Software Tools for Technology Transfer (STTT)*, 8(6):649 – 667, Oct. 2006.