pISSN:2349-5707 Volume:12, Issue 11, November-2025 eISSN:2349-5715

Architectural and Timing Analysis Frameworks for Deterministic Automotive Embedded Systems: A Comprehensive Study of End-to-End Delay, Model-Based Engineering, and Multicore Resource Management

Simona Reinhardt
Department of Computer Science, Technical University of Munich, Germany

ABSTRACT: The evolution of automotive systems into highly complex, distributed, and software-intensive platforms has significantly increased the demand for deterministic timing behavior and predictable system performance. Modern vehicles integrate numerous electronic control units (ECUs), sensors, and actuators, interconnected through heterogeneous communication networks and operating under stringent real-time constraints. This paper presents a comprehensive analytical study of architectural and timing analysis frameworks for automotive embedded systems, focusing on end-to-end delay analysis, model-based system engineering, and multicore resource management. Drawing upon existing literature, the study investigates compositional frameworks for delay calculation, cause-effect chain analysis, and model-driven methodologies such as EAST-ADL and MoVES. Additionally, it explores the challenges posed by multicore architectures, including memory bandwidth contention and scheduling complexities in mixed-criticality systems. The integration of fault-tolerant architectures, such as dual-core lockstep systems, is also examined to understand their impact on timing predictability and system reliability. The findings reveal that while existing frameworks provide valuable tools for early-stage analysis and system design, significant challenges remain in achieving accurate and scalable timing analysis in increasingly complex automotive systems. The paper highlights the need for unified methodologies that integrate architectural modeling, timing analysis, and resource management, and proposes future research directions aimed at enhancing predictability, scalability, and safety in next-generation automotive systems.

Keywords
Automotive Embedded Systems, End-to-End Delay Analysis, Model-Based Engineering, Multicore Systems, Real-Time Scheduling, Fault Tolerance, Predictability.

INTRODUCTION

The automotive industry has undergone a profound transformation over the past few decades, evolving from mechanically dominated systems to highly sophisticated cyber-physical platforms characterized by extensive software integration and distributed computing. Modern vehicles are equipped with advanced functionalities such as autonomous driving, adaptive cruise control, and real-time diagnostics, all of which rely on complex embedded systems that must operate with high levels of reliability and predictability. As the number of electronic control units (ECUs) and interconnected components continues to grow, the challenge of ensuring deterministic system behavior has become increasingly critical.

One of the central concerns in automotive embedded systems is the analysis and management of end-to-end timing behavior. End-to-end delay refers to the time taken for data to propagate through a sequence of interconnected components, often referred to as a cause-effect chain. These chains represent the flow of information from sensors to actuators, and their timing characteristics directly impact system performance and safety. For instance, delays in processing sensor data or transmitting control signals can lead to degraded performance or even catastrophic failures in safety-critical applications.

Early research in this domain introduced compositional frameworks for calculating end-to-end path delays under different path semantics, emphasizing the importance of modular analysis and abstraction (Feiertag et al., 2008). These frameworks laid the foundation for subsequent studies that focused on more detailed

and application-specific timing analysis techniques. For example, Becker et al. (2017) developed methods for analyzing cause-effect chains in automotive systems, highlighting the complexities introduced by asynchronous communication and varying execution rates.

The increasing complexity of automotive systems has also necessitated the adoption of model-based engineering approaches. Frameworks such as EAST-ADL provide a structured methodology for modeling system architecture, enabling designers to capture functional, logical, and technical aspects of the system in a unified manner (Kolagari et al., 2015). Similarly, model-driven methodologies like MoVES facilitate the integration of timing analysis into the design process, allowing for early identification of potential performance bottlenecks (Bucaioni et al., 2018).

Despite these advancements, several challenges remain in achieving accurate and scalable timing analysis. One of the key issues is the interaction between software and hardware components, particularly in multicore architectures. The introduction of multicore processors has significantly enhanced computational capabilities, but it has also introduced new sources of unpredictability, such as memory bandwidth contention and resource sharing. Studies have shown that these factors can significantly impact task execution times and overall system performance (Agrawal et al., 2018).

Another important aspect is the need to manage mixed-criticality systems, where tasks with different levels of importance share the same computational resources. Ensuring that high-criticality tasks meet their timing requirements while efficiently utilizing resources for lower-criticality tasks is a complex scheduling problem that has been the focus of extensive research (Awan et al., 2018a).

Furthermore, the integration of fault-tolerant architectures, such as dual-core lockstep systems, adds another layer of complexity to timing analysis. While these architectures enhance system reliability by providing redundancy and error detection mechanisms, they also introduce additional overheads that must be accounted for in timing models (Karim, 2023).

This paper aims to provide a comprehensive analysis of these challenges and the existing frameworks designed to address them. By synthesizing insights from the literature, the study seeks to identify gaps in current methodologies and propose directions for future research.

METHODOLOGY

The methodological approach adopted in this study is grounded in an extensive qualitative synthesis and theoretical integration of the referenced literature, focusing on three primary dimensions: end-to-end timing analysis, model-based architectural engineering, and multicore resource management in automotive embedded systems. Rather than relying on empirical experimentation, the research constructs a conceptual framework that systematically analyzes and interrelates these domains to provide a holistic understanding of deterministic system behavior.

The first stage of the methodology involves an in-depth examination of compositional timing analysis techniques. Foundational frameworks proposed for end-to-end delay calculation are analyzed in terms of their ability to handle different path semantics, including synchronous and asynchronous communication patterns. The study evaluates how these frameworks decompose complex systems into manageable components while preserving the accuracy of timing predictions. Particular attention is given to the propagation of delays across cause-effect chains and the challenges associated with modeling interactions between periodic and aperiodic tasks (Feiertag et al., 2008; Becker et al., 2017).

The second stage focuses on model-based engineering methodologies, with an emphasis on frameworks

such as EAST-ADL and MoVES. The analysis explores how these methodologies support the integration of timing constraints into system design, enabling early-stage validation of performance requirements. The study examines the hierarchical modeling structures employed in these frameworks, including abstraction layers that represent functional, logical, and technical architectures. Additionally, it evaluates the role of model refinement in improving the accuracy of timing analysis, particularly in legacy systems where existing constraints must be adapted to new requirements (Kolagari et al., 2015; Mubeen et al., 2016).

The third stage addresses the challenges associated with multicore architectures, particularly in the context of resource sharing and scheduling. The study analyzes various approaches to memory bandwidth regulation, including static partitioning and dynamic allocation strategies. It evaluates the impact of these approaches on system predictability, considering factors such as worst-case execution time and task interference. The analysis also explores scheduling techniques for mixed-criticality systems, assessing their ability to balance performance and safety requirements (Agrawal et al., 2018; Awan et al., 2018b).

The final stage integrates insights from fault-tolerant system design, focusing on dual-core lockstep architectures. The study examines how redundancy and synchronization mechanisms influence timing behavior and evaluates the trade-offs between reliability and performance. By combining these perspectives, the methodology provides a comprehensive framework for understanding the interplay between architectural design, timing analysis, and resource management in automotive embedded systems.

RESULTS

The analysis reveals that compositional frameworks for end-to-end timing analysis provide a robust foundation for understanding delay propagation in automotive systems. These frameworks enable the decomposition of complex systems into smaller components, facilitating modular analysis and improving scalability. However, the accuracy of these frameworks is highly dependent on the assumptions made לגבי task synchronization and communication patterns, which can vary significantly in real-world systems (Feiertag et al., 2008).

Cause-effect chain analysis emerges as a critical tool for evaluating system performance, particularly in applications involving sensor-to-actuator data flows. The findings indicate that incorporating detailed models of task interactions and communication delays can significantly enhance the accuracy of timing predictions. However, this level of detail also increases computational complexity, posing challenges for large-scale systems (Becker et al., 2017).

Model-based engineering frameworks such as EAST-ADL and MoVES are shown to be effective in integrating timing analysis into the design process. These frameworks enable early identification of potential performance issues, reducing the risk of costly redesigns at later stages. The use of hierarchical models and abstraction layers allows designers to manage system complexity while maintaining a clear representation of timing constraints (Kolagari et al., 2015; Bucaioni et al., 2018).

In the context of multicore systems, memory bandwidth contention is identified as a major source of unpredictability. Techniques such as bandwidth regulation and resource partitioning are found to mitigate these issues, but their effectiveness depends on the accuracy of workload characterization and the ability to adapt to dynamic system conditions (Agrawal et al., 2018).

The integration of fault-tolerant architectures enhances system reliability but introduces additional overheads that must be accounted for in timing analysis. Dual-core lockstep systems, in particular, provide robust error detection capabilities, but their impact on execution time and resource utilization requires

careful consideration (Karim, 2023).

DISCUSSION

The findings highlight the need for a holistic approach to timing analysis in automotive embedded systems, one that integrates architectural modeling, timing analysis, and resource management. While existing frameworks provide valuable tools for addressing individual aspects of the problem, there is a lack of unified methodologies that can handle the full complexity of modern systems.

One of the key challenges is the trade-off between accuracy and scalability in timing analysis. Detailed models provide more accurate predictions but increase computational complexity, making them impractical for large systems. This suggests a need for hybrid approaches that combine detailed analysis with abstraction techniques.

Another important issue is the dynamic nature of modern automotive systems, which often operate varying. Traditional timing analysis techniques, which on static assumptions, may not be sufficient to capture these dynamics. This highlights the need for adaptive analysis methods that can respond to changing system conditions.

The integration of multicore architectures and fault-tolerant systems further complicates the analysis, introducing new sources of unpredictability and overhead. Addressing these challenges requires advances in both hardware and software design, as well as improved coordination between different levels of system architecture.

Future research should focus on developing integrated frameworks that can seamlessly combine different analysis techniques, as well as tools that can automate the analysis process. Additionally, there is a need for standardized methodologies that can be applied across different platforms and applications.

CONCLUSION

This study has provided a comprehensive analysis of architectural and timing analysis frameworks for automotive embedded systems, highlighting the challenges and opportunities associated with achieving deterministic system behavior. The findings underscore the importance of integrating end-to-end timing analysis, model-based engineering, and multicore resource management to address the complexities of modern automotive systems.

While significant progress has been made in developing frameworks and methodologies, several challenges remain, particularly in achieving scalability, adaptability, and integration. Addressing these challenges will be critical for enabling the next generation of automotive systems, which will require even higher levels of performance, reliability, and safety.

REFERENCES

1. Agrawal, A., Mancuso, R., Pellizzoni, R., et al. Analysis of dynamic memory bandwidth regulation in multi-core real-time systems. IEEE Real-Time Systems Symposium, 2018.

2. Anandtech. NVIDIA Drive AGX Orin. 2019.

3. ARM. Arm Architecture Reference Manual Supplement. Memory System Resource Partitioning and

Monitoring (MPAM) for Armv8-A. 2022.

4. Awan, M. A., Bletsas, K., Souto, P. F., et al. Mixed-criticality scheduling with dynamic memory bandwidth regulation. IEEE RTCSA, 2018.

5. Awan, M. A., Souto, P. F., Bletsas, K., et al. Worst-case stall analysis for multicore architectures with two memory controllers. ECRTS, 2018.

6. Awan, M. A., Souto, P. F., Bletsas, K., et al. Memory bandwidth regulation for multiframe task sets. IEEE RTCSA, 2019.

7. 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, 80, 104–113, 2017.

8. Becker, M., Mubeen, S., Dasari, D., Behnam, M., Nolte, T. A generic framework facilitating early analysis of data propagation delays in multi-rate systems. IEEE RTCSA, 2017.

9. Brandenburg, B. Scheduling and locking in multiprocessor real-time operating systems. University of North Carolina, 2011.

10. Bucaioni, A., Addazi, L., Cicchetti, A., Ciccozzi, F., Eramo, R., Mubeen, S., Sjödin, M. MoVES: A model-driven methodology for vehicular embedded systems. IEEE Access, 6, 6424–6445, 2018.

11. Buttazzo, G., Bini, E. Optimal dimensioning of a constant bandwidth server. IEEE RTSS, 2006.

12. Buttazzo, G. Hard real-time computing systems: Predictable scheduling algorithms and applications. Springer, 2011.

13. Cinque, M., De Tommasi, G., Dubbioso, S., et al. Rpuguard: Real-time processing unit virtualization for mixed-criticality applications. EDCC, 2022.

14. Dagieu, N., Spyridakis, A., Raho, D. Memguard: A memory bandwidth management in mixed criticality virtualized systems. UBICOMM, 2016.

15. Farshchi, F., Huang, Q., Yun, H. Bandwidth regulation unit for real-time multicore processors. IEEE RTAS, 2020.

16. 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. CRTS Workshop, 2008.

17. Abdul Salam Abdul Karim. (2023). Fault-Tolerant Dual-Core Lockstep Architecture for Automotive Zonal Controllers Using NXP S32G Processors. International Journal of Intelligent Systems and Applications in Engineering, 11(11s), 877–885. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/7749

18. Kolagari, R. T., Chen, D., Lanusse, A., Librino, R., Lönn, H., Mahmud, N., Mraidha, C., Reiser, M.-O., Torchiaro, S., Tucci-Piergiovanni, S., Wägemann, T., Yakymets, N. Model-based analysis and engineering of automotive architectures with EAST-ADL: Revisited. International Journal of Conceptual Structures and Smart Applications, 3(2), 25–70, 2015.

19. Mubeen, S., Mäki-Turja, J., Sjödin, M. Support for end-to-end response-time and delay analysis in the

industrial tool suite: Issues, experiences and a case study. Computer Science and Information Systems, 10, 453–482, 2013.

20. Mubeen, S., Nolte, T., Lundbäck, J., Gålnander, M., Lundbäck, K.-L. Refining timing requirements in extended models of legacy vehicular embedded systems using early end-to-end timing analysis. Springer, 2016.

21. Mubeen, S., Nolte, T., Sjödin, M., Lundbäck, J., Lundbäck, K.-L. Supporting timing analysis of vehicular embedded systems through the refinement of timing constraints. Software and Systems Modeling, 39–69, 2019.

22. Thorngren, P. Experiences from EAST-ADL use. EAST-ADL Open Workshop, 2013.