Developing Vehicles for Future Transportation Systems: Challenges & Research Directions

David Garlan
Computer Science Dept.
Carnegie Mellon University

Bruce H. Krogh
Dept. of Elec. & Comp. Eng.
Carnegie Mellon University

Anthony Tsakiris
Research & Advanced Engineering
Ford Motor Company

In all transportation systems, there are increasingly stringent demands for vehicles in terms of features, performance, and fuel economy. To meet these requirements, manufacturers are developing more and more complicated vehicle systems, and do so in an increasingly competitive and economically challenging environment. For many of the solutions, we turn to systems that are software-intensive. The flexibility provided by electronics, software, and networking opens up many feature and performance possibilities. That same flexibility, however, sometimes results in highly inter-connected, complicated, and difficult-to-manage systems that fail to satisfy performance and business objectives.

To compete effectively, vehicle producers require software-intensive systems that (in addition to satisfying functional requirements) are robust, easy to develop, easy to implement across large ranges of vehicles, and easy to adapt as requirements and technologies change. To develop designs that meet these objectives, we need architecture specification and analysis techniques that help us make early and objective decisions about the properties, interactions, and performance of systems and their constituent software and hardware components. We also need the ability to support reusable architectural frameworks and patterns within the context of automotive architectural specifications.

Current Limitations

Current tools for developing transportation vehicles are insufficient to meet current challenges. Resources—development time, people, finances—are severally limited, making it difficult to meet the demands for more and better products, that cost less to develop and produce, and are brought to market faster. Important inadequacies in current development tools include:

- features for managing variability of components within and across similar products;
- abstractions and operations for managing complexity of systems with large numbers of components, complex interactions, and mixtures of physical and computational elements; and
- effective means for expressing complex and distributed systems for communication among

many team members.

- features for managing globally-distributed development responsibility, where significant pieces of software are developed by both OEMs and suppliers using different processes and under different timing and reuse constraints.

Research Challenges

Research is needed on fundamental concepts and methods that will address the above limitations for transportation vehicles. We believe many of the tool needs share common elements across specific transportation system domains. These crosscutting challenges include the following:

- Languages (broadly construed) for specifying software architecture for embedded, real-time, closed-loop control systems, including non-functional requirements. UML, data flow diagrams, control block diagrams, etc. are all good, but none seems to fully cover what is needed in this area. Analysis of systems in which both physical and computational properties must be considered. Additionally, compositional analysis capability is needed, whereby different quality dimensions (performance, reliability, etc.) can be investigated in isolation, but later combined to provide a unified model.
- Methodologies for migrating from legacy systems and components to new, better-structured systems. Vehicle development is never a clean-sheet exercise; there are always many constraints imposed by legacy subsystems that limit restructuring.
- Computer-aided tools for analysis and design of embedded control systems at the architectural level with the ability to enforce architectural integrity constraints across all products in a family. Tools must be “easy” to use (an example of “easy” is what Simulink and Stateflow have done for control algorithm development).

Promising Directions

Several initiatives have begun to address some of the above challenges. From the industry perspective, standards such as AADL (SAE AS5506), IEEE Std. 1471-2000, and ISO/IEC 42010 (currently being drafted) are important foundations for supporting commonalities among tools and platforms. Important research directions include:

- Formal representations of these frameworks, where conformance can be automatically checked and enforced.

- Formal architectural modeling approaches – raising the level of design abstraction, providing a common model for integration of quality dimensions, and supporting enforcement of integrity constraints.
  - Use of views to extract partial models.
  - New ways to combine physical and cyber behavioral models.
- Model-based self-healing systems – in which run-time models are used to detect problems and adapt systems dynamically
- Standardized analysis methods and tools. – permitting interchanging and communication across organizations

Milestones

We propose the following goals for a broad research initiative in development tools for transportation vehicles:

5 years: standard(s) for documenting software architecture of large, complex automotive systems, together with tools that can check for compliance

10 years: integrated analysis tools that allow multi-dimensional design and integration.

20 years: long-lived system abstractions that work for many vehicle technologies (conventional vehicles, hybrid electric vehicles, electric vehicles, fuel cell vehicles, ever-expanding active safety functionality, vehicle-to-vehicle functionality, vehicle-to-infrastructure functionality)

David Garlan is a Professor in the School of Computer Science at Carnegie Mellon University where he leads several research projects and is the Director of Professional Software Engineering Programs. His research interests include software architecture, pervasive computing, self-healing systems, applied formal methods, and software development environments. He can be contacted at garlan@cs.cmu.edu.

Bruce H. Krogh is a Professor in the Department of Electrical and Computer Engineering at Carnegie Mellon University. His current research interests include synthesis and verification of embedded control software, distributed sensing and control strategies, and the integration of control theory with formal methods in computer science. He can be contacted at krogh@ece.cmu.edu.

Anthony Tsakiris is a technical specialist in vehicle controls architecture in Research and Advanced Engineering at Ford Motor Company. He works on software controls for hybrid electric and fuel cell electric vehicles, and on related process improvement. He has a Bachelor of Science degree from The University of Michigan and a Master of Science degree from Stanford University, both in mechanical engineering. He can be contacted via email at atsakiri@ford.com.