Design Automation Challenges in Automotive CPS

Sayan Mitra
mitras@illinois.edu

In principle, best-effort technologies can be used for building each individual automotive cyber-physical system (CPS) from the ground-up, through careful design, testing, and verification. Each such undertaking, however, is technically challenging, error-prone, and expensive. Since many of these systems share common challenges, employ common design patterns, and verification principles, it is expected that generic software tools for automating design, testing, and verification can alleviate these challenges. In this position paper, we reflect on the key barriers we confront in the near term (next 5 years) towards developing such software tools and we present some ideas on overcoming them. We focus on verification, though the points we make apply to design and testing to a large extent.

Benchmarks

For making sustained progress, we need to be able to measure progress. For measuring effectiveness of verification tools, we need standardized benchmarks. Benchmark repositories have been a major driver of research and development in other areas in engineering such as the computer architecture, circuits, Electronic Design Automation (EDA), hardware verification, and automatic theorem proving. Unfortunately, a standardized open repository of benchmarks for automotive CPS is currently not available. In addition to enabling tool comparisons, by connecting graduate student's research to industry, benchmarks will enable the creation of an ecosystem of academia, automakers, and start-up companies, in which each play a role and the automotive CPS community flourishes.

A good benchmark repository (for verification) represents typical models and exposes the worst-case models. It is often difficult for companies to disclose real models owing to IP and security related issues. This can be overcome, for example, by creating an organization, such as the Standard Performance Evaluation Corporation (SPEC), for redacting real-world models and presenting them in a standard format, say Simulink/Stateflow. Alternatively, we could create an internship program whereby students interact with automotive engineers expressly for accomplishing the same goal.

Abstraction and Model Reduction

The state-space explosion problem has been a steep barrier in automated analysis of cyber-physical system models. To overcome this problem we have to construct of approximations or abstractions of such models. Over the past two decades, several approaches have been proposed for automatic abstraction including predicate abstraction, timed and hybrid abstractions, approximate abstractions, and counter-example guided abstraction refinement. Another promising direction of research is parametric verification, which can potentially circumvent the state-space explosion problem by (automatically) establishing that verification of a large number of interacting components can be reduced to the verification of a small number of abstract components. Finally, there has been a steady flow of ideas from control theory, such as, barrier certificates, small gain theorems, and dwell time theorems, which are supported by powerful optimization-based tools (e.g., sum-of-squares, linear matrix inequalities), and therefore, can be used effectively for solving verification problems. By pursuing all these directions and by creating tools that embody these techniques, in the near future, it should become possible to automatically verify models of moderately complex automotive components at a compelling level of detail.

Middleware for Verifiability in the Large

An automotive CPS is a distributed system—a variety of sensors, actuators, and computers interact over a communication bus. As the number of components in automotive CPS grow, they will inherit more of the complexities of distributed systems: failures, concurrency, and asynchrony. Despite the advances made in the areas of formal methods and hybrid systems, most verification techniques are designed for non-distributed systems. In fact, automotive CPS with failures, human factors, message delays, and unreliable communication are unlikely to be amenable to automatic analysis without a suitable theory for composition. Furthermore, existing verification approaches are unaware of the semantics of the underlying communication and computation layers of the CPS.

In a way, this is a missed opportunity because such "agnostic verification" confronts all the complexities (message delays, failures, etc.) that are encountered during design, and possibly managed in ways which can also aid verification. Consider a computer that receives message from N channels each of which can queue up to L messages. Without knowledge of the channel semantics, the system model will have $O(NL)$ non-deterministic transitions—one for each message being delivered. In contrast, if the middleware guarantees FIFO delivery, the number of transitions reduce to $O(N)$. A stronger semantics, say, total ordering and bounded latency can further reduce this to $O(1)$ transitions. Thus knowledge of the semantics of the underlying channel dramatically simplify the model, and hence, verification. This motivates the need for a middleware which not only facilitates implementation of applications, but also aids verification.

In order to develop such a middleware, we propose the development of a stabilizing group communication service (GCS) consisting of: a *reliable multicast service* that provides sensors, actuators, and computers (group members) with precise guarantees about message delivery, and a *group membership service* that provides group members with a consistent view of the set of nonfaulty members. A key component of the latter is a *failure detector (FD)* service. The FD service could use usual strategies such as heartbeats, and techniques that use reliable physical signals in detecting cyber components failures (e.g., odometer to detect GPS failures). The GCS will be stabilizing in the sense that a certain invariant property S will be preserved always (even with failures), and a (possibly stronger) invariant property G will be satisfied some bounded time after the failures, recoveries and dynamic changes cease. Typically S will be used for proving safety and G for proving progress of the higher-level application. Stabilization enables us to compositionally verify complex systems by establishing the stabilization of independent elementary services (e.g., GMS, FD, etc.), and with semantics-aware composition theory for message reordering tolerance, delay tolerance, and delay insensitivity that decomposes a verification task into a set of smaller tasks.

Conclusion

We discussed several key challenges and potential directions for overcoming them in developing effective software tools for design and verification of automotive CPS. Existing abstraction-based verification techniques have the potential to mature into tools that could verify automotive components of moderate complexity. For distributed systems, we propose the development of a group communication middleware with well-defined semantics, and the supporting theory that aids compositional verification. We believe that it is critical for the community to develop a system for generating a repository of standardized benchmarks for evaluating and informing the research in automated design and verification of automotive CPS.

Bio

Sayan Mitra is an Assistant Professor of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign since 2008. His research interests are in verification of cyber-physical systems, distributed algorithms, and automotive applications. Prior to joining Illinois, he was a post-doctoral fellow at the Center for Mathematics of Information of California Institute of Technology for a year. There he participated in a project related to verification of CalTech's autonomous vehicle for DARPA Urban Challenge, and continues work on automotive systems in collaboration with John Deere & Co. He earned his Ph.D. from MIT in 2007 and received National Science Foundation's CAREER award in 2011. Phone: 217 333 7824, Email: mitras@illinois.edu.