Position Paper: NSF CPS Transportation Workshop Jan. 2014

Formal Synthesis of Discrete Control Logic for Safe Transportation Systems

submitted by

Stéphane Lafortune, University of Michigan and *Richard Hill*, University of Detroit Mercy

December 12, 2013

Motivation

Modern vehicles include an extensive amount of discrete logic, implemented in software, that is needed to control and safeguard various aspects of a vehicle's operation: braking systems, steering systems, occupant safety systems, etc. Currently, most discrete logic design is performed heuristically in a decentralized manner. Problems that arise due to unexpected interactions among different systems are generally identified during vehicle-level simulation, or worse, during tests on actual vehicles. At this point, debugging an underlying flaw is much more difficult and the resulting fix is generally more complicated and expensive to implement. The earlier in the design process a problem can be identified, the better. It is further very likely that errors will fail to be captured by the testing process altogether. As vehicle systems are becoming increasingly complex and design cycles increasingly short, current techniques for verification and validation of the discrete logic have become insufficient to produce the performance and reliability that is required.

In addition to vehicle-level challenges for discrete logic synthesis, the development and deployment of new capabilities in intelligent transportation systems is posing similar challenges at the system level, i.e., at the level of a set of interacting vehicles. Several types of new active safety systems, for instance, involve discrete logic for coordinating a set of vehicles with autonomy capability, either in a fully distributed manner, or in tandem with a coordinating controller that is part of the transportation infrastructure. Again, the fundamental challenge involves the efficient synthesis of distributed software modules that provably satisfy the given safety and resiliency requirements, while avoiding undesirable interactions that may lead to deadlock or other faulty behavior.

While formal software verification techniques based on model-checking, for instance, are beginning to be employed in industry to mitigate the above-mentioned challenges in verification and validation, we submit that the emphasis has to be beyond verification and instead on the incorporation of *reactive synthesis* techniques at the outset of the discrete logic design process, both at the vehicle level and at the system level in the context of intelligent transportation systems. Reactive synthesis refers to the formal synthesis of discrete logic that is provably correct with respect to given specifications and that operates in an open environment subject to exogenous events, as is the case in automotive and transportation systems.

The *Supervisory Control* paradigm for the formal synthesis of correct-by-construction discrete control logic is a reactive synthesis methodology that has been developed in control engineering for dynamical systems modeled as discrete event systems. This discrete-event framework has the ability to model, verify, and synthesize autonomous and semi-autonomous systems, including their interaction with unpredictable and partially unknown environments, so that the system is guaranteed to meet specified discrete requirements without the use of exhaustive testing programs. Other reactive synthesis techniques have been developed in theoretical computer science, specifically for discrete transition systems subject to specifications expressed in some type of temporal logic. All of the above techniques can be applied at the component level as well as at the system level, and can be coupled with existing or modified optimization techniques to additionally achieve efficient global system behavior. Working with discrete-event models that are built by abstraction from the continuous dynamics of the underlying system is arguably the most effective way of tackling discrete logic synthesis problems for many classes of complex cyber-physical systems, such as vehicular transportation systems, that are subject to safety, liveness, and resiliency requirements. On the one hand, safety and liveness can be handled in the synthesis of discrete-event controllers, by the suitable mapping of safe and unsafe continuous system states to corresponding safe and unsafe discrete states during the

abstraction step, and by the expression of liveness properties in terms of eventual reachability of certain discrete states. On the other hand, resiliency can be analyzed by studying the diagnosability properties of discrete-event abstractions, which will tell us if and when the most common types of faults of system components will be detected by model-based inferencing driven by run-time observations.

Research Challenges

Research challenges regarding the application of reactive synthesis techniques, from control engineering or from theoretical computer science, to transportation cyber-physical systems abound. First, building discrete-event models at the right level of detail from the underlying continuous or hybrid dynamics of a single vehicle or of a set of vehicles requires powerful abstraction techniques. In this regard, it is imperative to exploit any type of symmetry that may exist in the underlying system. This is particularly relevant at the system level, since different vehicles may have similar abstracted models. A related issue is that the level of detail in the abstracted model should be commensurate with the formal specifications imposed on the system, regarding safety, liveness, and resiliency. Quite often, the task of taking a natural language description of a specification and formalizing it as an automaton or a temporal logic formula is itself very challenging. Second, reactive synthesis algorithms, either from supervisory control theory or from computer science, must be adapted to the distributed nature of the system. At the vehicle level, we have a set of interacting systems within a single vehicle; at the system level, we have a set of interacting vehicles. Moreover, the synthesis algorithms must account for measurement uncertainty, as key vehicle variables such as position or velocity are typically obtained by noisy sensors. Finally, for safety reasons, the synthesized discrete logic should always allow graceful degradation to a fail-safe mode of operation when faults or unexpected events occur. All of these requirements must be satisfied by algorithmic synthesis techniques that must scale to transportation systems of increased complexity, both at the vehicle level and at the system level.

In summary, the research agenda for autonomous or semi-autonomous vehicles operating in intelligent transportation systems is rich in problems of discrete logic synthesis for the various software modules that will be included in future vehicles and in the transportation architecture. These problems will require synergistic approaches that leverage advances in control engineering, reactive synthesis, and verification and validation.

Authors

Stéphane Lafortune: EECS Dept., University of Michigan; stephane@umich.edu; 734-763-0591; fax: 734-763-8041.
`http://web.eecs.umich.edu/~stephane/`

Richard Hill: ME Dept., University of Detroit Mercy; hillrc@udmercy.edu; 313-578-0428; fax: 313-578-0428.
`http://hillrc.faculty.udmercy.edu/`