Ensuring the Safety of Transportation Cyber-Physical Systems

Linh T.X. Phan Insup Lee Oleg Sokolsky
University of Pennsylvania
{linhphan, lee, sokolsky}@cis.upenn.edu

Christopher Gill Chenyang Lu
Washington University in St. Louis
{cdgill, lu}@cse.wustl.edu

1 Introduction

Modern transportation cyber-physical systems (CPS) are increasingly complex and autonomous. They are built through extensive collaboration between a system integrator and a multitude of suppliers, and such collaborative development calls for modular designs that enable different teams to work on different parts of a system independently. In addition, most mass-produced systems, such as cars and airplanes, are designed based on product-line architectures that allow easy customization to changing customer needs. As a result, systems are built as collections of cyber-physical components or *features*: each feature provides a particular functionality, such as adaptive cruise control in automotive systems, that appears in several different product lines. These independently-developed features co-exist on a shared cyber-physical platform, and may co-control the vehicle's actuators or even autonomously drive the vehicle.

Due to this increase in complexity and autonomy, the features of transportation CPS are also becoming increasingly safety-critical. Moreover, they have different safety criticality levels (e.g., level A through level D according to the ISO 26262 standard) that require different levels of safety assurance, and their criticality levels can also change dynamically depending on the environment conditions (e.g., the road condition) or the current state of the system (e.g., the speed of the vehicle). Thus, it is important that the implementation of these features guarantees their dynamic safety requirements. Further, the integration of these features into a cyber-physical system must also ensure the overall safety of the system.

Although there exist a number of design and analysis frameworks for CPS, they focus solely on the cyber layer, while ignoring or making implicit assumptions about the physical and cyber-physical semantics of the features and their interactions. As a result, failures can occur within the system when these hidden assumptions are violated. To guarantee the safety of the system, it is therefore critical to extend these theories beyond cyber concerns.

2 Research Challenges

An important step towards safe modular design and analysis of transportation CPS is to develop **safety-aware component models** for CPS that can encapsulate details of a feature's implementation while exposing only the *safety interface* for the feature it encloses. Such a model needs to capture both the cyber and the physical aspects, as well as their interactions, of a feature – including e.g., the functionality, the timing constraints and resource requirements, the dynamics of the physical system under control, the control algorithm, safety-criticality levels of tasks, and safety goals of the feature under different operating conditions. In addition, the safety interface of a feature should specify all possible interactions of the feature with its environment and the constraints on these interactions that are necessary to ensure the safety of the feature when it is integrated into the cyber-physical system. This is highly challenging, since it is not always possible or feasible to identify implicit dependencies between the feature and the environment, due to the

uncertainty and the complexity of the physical environment and the user's behavior (e.g., the reactions of the driver in a vehicle under different driving situations).

Another critical challenge is to **detect unsafe interactions among features** and to **manage their interactions** during the system integration. Undesirable interactions between features on the same cyber-physical platform can arise in several ways, such as through shared data and variables, through shared actuators and sensors, through the physical environment, and through computational and communication resource sharing. To illustrate this, let us consider three common features of an automotive system: collision avoidance (CA), automated lane-centering control (LC), and adaptive cruise control (ACC). An undesirable interaction between the CA and the LC via shared variables might occur when these features assign conflicting torque values, or request an increase on the torque value one immediately after the other, which could result in over-steering. An unsafe interaction between these two features might also occur via shared actuators, e.g., when they control the steering shaft simultaneously, creating a deadlock or letting the wrong feature take control. Similarly, an unsafe interaction through the physical environment might occur between the ACC and the CA when the former requests an increase in speed while the latter simultaneously requests a sharp turn, which could cause the vehicle to roll over. Finally, unsafe interactions via platform resource sharing might arise, e.g., when tasks of the ACC consume too much communication bandwidth and delay an urgent task of the CA, which could lead to a collision. Alternatively, a low-criticality task of the former might consume too much execution time and fail to leave enough time for the high-criticality tasks of the latter to meet their deadlines.

There exist several formalisms that can serve as building blocks for feature interaction analysis and management. For instance, safe shared-variable interactions (e.g., non-conflicting assignments to the shared variable representing the torque value) can be formally described, using temporal logic, as properties of the variables – which can then be verified using formal verification. Similarly, unsafe interactions via a shared CPU can be detected and eliminated using timing analysis and real-time scheduling techniques. However, addressing the above challenge in the general case requires new approaches that span "across all layers" of the CPS. Below, we identify some of the open research questions towards this goal.

- *How to derive new safety requirements during the integration of features?* Because of the co-execution of the features, it is highly likely that their safety-criticality levels are changed and new safety goals are required (e.g., due to new hazard scenarios that arise in the composition of the features) but could not be identified during the safety analysis of each individual feature, such as the interaction between the ACC and the CA via the physical environment that causes the vehicle to roll over described earlier. Therefore, methods for automatically generating all such new safety requirements for the composition and for extracting from the new requirements the concrete formal properties that each feature must guarantee are needed to ensure the safety of the integration.

- *How to analyze the effect of resource interferences?* Features in transportation CPS often share many different types of platform resources, such as processor cycles, memory devices, input/output devices, and serial communication controllers. Hence, extensions of existing timing analysis are needed to analyze not only direct interference between features through single-resource sharing (e.g., the same CPU) but also indirectly interference through inter-relationships between multiple resources (e.g., CPU and memory). At the same time, new safety-aware adaptive resource management techniques are needed to enforce safe feature interactions via resource sharing and to ensure the dynamic safety requirements of the features.

- *How to tackle the coupling between control, safety, and resource aspects of the features?* As an example, the safety of the CA feature depends on the stability and control quality of its control algorithm, whose behavior depends on the resource sharing semantics of the corresponding control tasks, which in turn is driven by the safety-criticality levels of the tasks and hence, the safety requirements of

the feature. New approaches that integrate control theories, safety assurance techniques, and timing analysis methods are crucial to efficiently and accurately analyze the effects of this complex coupling.

- *How to detect and analyze implicit dependencies among features?* Due to the complexity and uncertainty of the physical environment and the human behavior, it is not feasible (or even possible) to statically detect all potentially unsafe interactions. As a result, it is necessary to develop run-time detection and recovery techniques that can be used to continuously monitor the system during its execution, check for any potential unsafe interactions, and respond accordingly to ensure the system maintains its safe behavior. In addition, methods for automatic refinements of the safety interfaces and feature interactions during run time are crucial to keep up with the growing autonomy of today's and the next generation transportation CPS.

3 Conclusion

In this position paper, we have identified a number of important challenges towards ensuring operational safety of transportation cyber-physical systems. As these systems continue to grow in complexity and autonomy, new safety-aware modeling, interaction analysis, and safety enforcement approaches that capture features' execution semantics across all layers of the cyber-physical systems are needed to improve the safety of the current and the next generation of such systems.