Safe Collaborative Driving Systems

Nick Maxemchuk
Columbia University

The current generation of intelligent vehicles monitor their environment and warn the driver of impending danger or operate braking, advanced cruise control, and lane maintenance systems. Experimental vehicles are being constructed that can operate without a driver. The next generation of truly intelligent vehicles will operate cooperatively, rather than autonomously. They will cooperate with other vehicles, to share sensor readings and coordinate maneuvers, such as braking, cruise control and lane changes; they will cooperate with the infrastructure, to plan least energy routes and control traffic signals; and, they will cooperate with the driver, to make the operation of the vehicle safer.

The protocols that define the collaborative operations are complex. They interact with the physical world in multiple ways and are time critical. They must operate with older vehicles that do not have their capabilities and with multiple generations of the same protocol, created by many manufacturers. Errors in the design or implementation of these protocols can result in the loss of life. It is irresponsible to allow these vehicles on public roadways without creating engineering and testing procedures to guarantee their safety.

We must emphasize engineering rather than implementation. Engineers are trained to predict how a system will operate before the complete system is assembled. We didn't send the first men to the moon then try to figure out how to get them back. The systems were carefully engineered and the components of the system were completely tested and improved before the first human was sent into space.

The intelligent vehicles that we are constructing can endanger many more people than the first moon shot. The engineering task is more difficult because many competing organizations are designing the vehicles that must cooperate. We should take greater care in engineering these vehicles than we did with the space program. Intelligent vehicles are more than computers and cannot be simply rebooted after a crash. We cannot allow adventurers to put vehicles on public roadways without doing everything possible to guarantee our safety.

There are three steps that we must take to improve the safety of intelligent vehicles:

1) We need an architecture that allows us to decompose the complex problem into more manageable pieces and also creates a framework for a standards process.
2) We need better ways to deal with the time critical nature of the systems.
3) We must implement a verification and testing procedure that recognizes the competitive environment.

1) The architecture should have properties that are similar to those that have evolved in the layered architecture that is used in communications systems. One

architecture that we are investigating has a dimension for each interaction with the physical world, and a dimension for the applications. Each dimension is organized as a stack, similar to the stack architecture used in communications. Each stack separates the physical properties of the dimension from the logic in an intelligent application. Each layer of each stack provides well defined services over well defined interfaces to the layers above it.

The layered architecture simplifies the verification of systems by breaking the problem into smaller pieces. When we verify a component of the architecture, we prove that a layer provides a service, assuming that the services provided by lower layers have been verified. In a merge protocol we can prove that the merge protocol is safe assuming that an intelligent cruise control protocol, in a lower layer, can create and maintain a gap that is specified.

The architecture allows us to change the operation of any layer without changing the other layers. The communications stack has made it possible to change the transmission technologies in the Internet without changing the email application. In CPS systems we may even be able to reuse entire stacks in different applications. For instance, a communications stack and timing stack may provide services that are the same in many collaborative CPS systems.

2) Temporal logic has been used to simplify the verification of protocols by reducing the number of sequences of events that we must consider. In modern systems we can do more by using synchronized clocks. Accurate clocks, obtained from GPS, are especially appropriate in automotive systems that already use GPS for other applications. When a GPS signal isn't available, the clocks can be maintained with accurate crystal oscillators, or the NTP or PTP protocols can be used to synchronize clocks in adjacent vehicles.

Synchronized clocks are particularly useful in multi-party systems that initiate operations by sending messages over an unreliable communications channel. The message may require a different number of transmission attempts to reach the different participants, and result in different execution sequences that cannot be reduced by using temporal logic. Synchronized clocks can reduce the number of execution sequences by specifying the time that an operation is performed, rather than initiating timers at uncertain starting times.

3) Intelligent automobiles must be able to collaborate safely with all of the other intelligent automobiles on a highway. Testing all combinations of implementations, for all of the automobile manufacturers, all of their different models, and all of the different model years, can result in very large numbers of tests. We used an alternative strategy to test telephone equipment after the telephone network was opened to competition. This same strategy should be applied to the intelligent vehicles. The strategy decomposes the problem into the verification of a model and testing each implementation against the model. We create an unambiguous, formal model of a protocol and verify that the model provides a safe service. We then generate a black box test sequence to determine if an implementation correctly and

completely implements the model. The black box test sequence applies a set of inputs to the implementation and observes the outputs. If an implementation passes the tests, it will interoperate with all of the other implementations that pass the test. We have successfully generated tests for communications equipment that is not heavily dependent on time. We are investigating applying the test generation procedures to CPS systems that are time critical, but use synchronized clocks and store the deadlines for all of the participants in a replicated memory.