Carnegie Mellon

Electrical & Computer ENGINEERING

Pittsburgh, PA 15213-3890
ph: 412.268.5225
koopman@ece.cmu.edu • www.ece.cmu.edu/~koopman

**Philip Koopman, Ph.D.**
Associate Professor: ECE, RI, ISR

October 6, 2016 **Docket NHTSA-2016-0090**

Subj: Response to Federal Automated Vehicles Policy: accelerating the next revolution in roadway safety, September 2016

*This document is the sole opinion of the author, and does not necessarily reflect the University's views.*

It is very encouraging to see DOT & NHTSA taking a comprehensive look at the critical topic of Highly Autonomous Vehicle (HAV) safety. Overall, the report does a good job at proposing a baseline for discussing how an appropriate level of safety can be achieved in this quickly evolving area. I want to acknowledge the significant efforts of the authors of the report, and especially the contributions that presumably came from the staff of the Volpe Center.

**My Background:**

I have been involved in autonomous vehicle safety for 20 years. My first experience was as the computer dependability advisor to the Carnegie Mellon University team participating in the Automated Highway System (AHS) program. I have since worked on this topic at the National Robotics Engineering Center with funded projects on autonomous vehicle safety and robotic system dependability. I have experience with software safety, embedded computing systems, computer system dependability, and expert testimony regarding automotive software safety. In addition to my university experience, I have worked in industry as an automotive component designer. I am co-founder of Edge Case Research LLC, which emphasizes improving embedded software quality, software stress testing, and assuring autonomous vehicle safety.

**Topics:**

While the policy statement is an excellent start at dealing with this difficult area, I recommend policy improvements in the following areas:

1. Requiring a safety argument that deals with the challenges of validating machine learning
2. Requiring transparent independence in safety assessment
3. Triggering safety reassessment based on safety integrity, rather than "significant" functionality
4. Requiring assessment of changes that can compromise the triggering of fall-back strategies
5. Characterizing what "reasonable" might mean regarding anticipation of exceptional scenarios
6. Assuring the integrity of data that is likely to be used for crash investigations
7. Diagnostics that encompass non-collision failures of components and end-of-life reliability loss
8. More uniform codification of traffic rule exceptions
9. Ensuring the safety of driver-takeover strategies for SAE Level 2 systems

**Topic #1: System Safety and Machine Learning Validation**

**Recommendation: A safety argument should be required that explicitly addresses the special challenges of validating machine learning (or that demonstrates that traditional software technology can be relied upon despite arbitrarily bad failures of the machine learning system components).**

*Discussion (see policy pages 20-21; 27-28):*

It is good to see an endorsement of software safety standards in the proposed policy, and I strongly support such an endorsement. Appropriate safety standards should be followed in creating safe vehicles.

However, a critical gap in those standards is that there does not seem to be a clear and obvious way to trace training data for Machine Learning (ML) to requirements and design as is done when validating conventional software. Machine learning is commonly used in autonomy systems to perform tasks such as object identification. While it is possible that a satisfactory safety argument can be made for a system that employs ML in a safety-critical role, such an undertaking seems to be uncommon and might be impractical in the near term for a system as complex as a Highly Autonomous Vehicle (HAV). Therefore, the policy should deal with this issue explicitly as an emerging technology area and in fact an area of potentially very significant risk to HAV safety in general.

An example of a potential statement dealing with this area might be as follows. Mapping Machine Learning (ML)-based systems to traditional safety standards is challenging because the training data set does not conform to traditional expectations of software requirements and design. The safety assessment should include a safety argument that justifies the completeness and correctness of the ML data set, training process, and validation process used. This argument should include at least the topics of:

- Representativeness of data for the Operational Design Domain (ODD)
- Adequacy of training and testing methodology (e.g., to avoid over-fitting)
- Representativeness of the validation testing environment, and statistical analysis of whether validation via testing (including simulation) is sufficient to demonstrate required safety goals
- Analysis of brittleness when encountering unusual or novel data
- Validation of an appropriate capability (tied to SAE autonomy level) of the system to recognize when it is operating in conditions outside the intended ODD, recognize that it has encountered an anomalous situation it is unequipped to deal with, or otherwise recognize that the assumptions made in the safety assessment have been violated

One could alternatively envision a system in which non-ML software assures safety while the ML software provides enhanced functionality, in which case ML safety assurance might not be required.

It is extremely important to note that this comment is NOT meant to say that the ISO 26262 standard does not apply. Rather, it is essential that ISO 26262-style safety engineering be performed. Within that context, ML data sets either need to be credibly mapped into the standard's framework, or something additional must be done beyond ISO 26262 for ML validation.

*Selected Additional References:*

*Koopman, P. and Wagner, M., "Challenges in Autonomous Vehicle Testing and Validation," SAE Int. J. Trans. Safety 4(1):2016, doi:10.4271/2016-01-0128*

*Kalra, N. and Paddock, S., "Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?" Document RR-1478-RC, Rand Corporation, doi:10.7249/RR1478*

**Topic #2: Safety Assessment Independence**

**Recommendation: Transparent independence should be required to ensure the integrity of the safety assessment process.**

*Discussion (see policy pages 31, 78):*

The requirement for independence in testing must be strengthened. The key phrase currently reads: "Testing may be performed by manufacturers and suppliers but could also be performed by an independent third party." In other safety critical system domains, the requirement for independence is mandatory, not optional. In practice, *independence* of validation (including both some portion of validation testing and also ensuring conformance to appropriate software safety practices) is essential.

Events in recent years with various automotive safety scandals have made it clear that automotive companies are subject to failure when they attempt to internally self-police safety – just like any other industry would be. It is high time that the automotive industry joined the rest of the world in being required to have a transparent method of independent safety assurance. Note that strictly speaking this does not require the government to directly inspect their systems, nor does it require hiring a third party to do so. In practice those are the two common approaches, and in fact even today third parties routinely assess safety in at least some automotive components. However, external safety assessment is currently optional for car companies and has clearly not been carried out in some situations where it arguably should have been done.

It can be up to the industry to propose a credible system that provides a transparent approach to independence without needing to divulge proprietary information. The requirement should at least be to a credible mechanism. That mechanism must assure the integrity and independence of the safety assurance process despite pressure to acquiesce to unsafe practices even from top management of the organization. The onus should be on the vehicle and equipment makers to demonstrate sufficient independence as part of being able to claim safety. This should be relatively straightforward to accomplish for organizations who already have a strong safety culture.

Page 78 of the policy states that NHTSA might require or practice additional oversight. Whether by the government or some other entity that can establish independence, some general idea of independent oversight should be promoted from a "tool" considered for use to instead being part of the baseline policy statement. In other words, independence ought to be a "shall" or at least a "should" in the policy statement rather than noted in an addendum. It is noteworthy that page 78 says that oversight "would bring NHTSA's practices more into line with those other agencies … use to ensure the safety of software-driven products and systems." Clearly the authors of this document understand this point, and what remains is a matter of deciding to actually require independence.

**Topic #3: Triggering Safety Reassessment Via ASILs Rather Than "Significant" Changes**

**Recommendation: Reassessment of safety (e.g., submission of a new Safety Assessment letter) should be triggered every time any change is made to a safety critical system component, rather than using a subjective threshold of "significant" system-level changes.**

*Discussion (see policy pages 16-17):*

The motivation for submitting a new Safety Assessment letter only when there is a "significant update" is understandable to avoid an undue burden on those making seemingly trivial updates. However, in practice it can be difficult to distinguish a "significant" update from a "minor" update. Moreover, since submitters will presumably be inclined to minimize the number of Safety Assessment letters they have to prepare, it will be all too easy to abuse the "significant" threshold and characterize even substantial changes as "minor." However, even a "minor" update can cause significant problems. It only takes one line of bad code in the wrong place to compromise an entire computing system. One recent example is the widespread "Heartbleed" vulnerability, caused by a single line of bad code. Other severe system failures are routinely traced down to a seemingly small change or defect in code.

The proposed policy seems to casts significance in terms of the system capabilities (e.g., a change is "significant" when a new feature is added). However, an implementation change to the software that does not alter the intended feature set could be in fact be significant. Even a seemingly small bug fix can easily cause a new, different bug to appear that is potentially worse than the bug that was supposed to be fixed.

To illustrate that the proposed threshold could be easily abused, purely as an illustrative example, it seems that under the current wording someone could rewrite every line of code in the vehicle from scratch, use a different operating system, and run it all on a new and unproven hardware platform, and still claim it is not a "significant" change because the intended features did not change. Due to the possible inclination of submitters to mis-categorize changes, it is likely that an approach based on a subjective interpretation of which changes are "significant" will not be viable over the long term, and will instead lead to a huge loophole in the proposed policy.

Instead, the policy should promote the use of a time-tested safety approach such as the one that follows:

- Every change, *no matter how minor*, to a high-safety-integrity subsystem should trigger a new safety evaluation (a Safety Assessment letter in terms of the current policy wording). This includes "bug fixes" to safety critical functions. If the change is truly minor, then the effort required should simply be a very small change to the existing safety argument, or perhaps even an annotation explaining why this change is not relevant to the existing safety argument. In other words, the safety assessment effort should be proportional to the size and scope of the change.
- Submitters should be encouraged to modularize safety arguments by exploiting strong isolation mechanisms between system modules. That strong isolation permits them to argue that changes to other parts of the system cannot undermine system safety of a module being assessed. At the system level, this means that the safety assessment of each module is relatively independent, and need not be revisited when other (especially non-safety) modules have been

changed. This type of argument has been routine for decades in other safety critical system domains, and is already common practice.

- System designers can then partition their system so a safety assessment can be made on a modular basis, and segregate unlikely-to-change functions from rapidly changing functions to reduce their safety assessment burden. The concept of "significant" then maps on to how many modules need to be re-assessed and the scope of change to existing safety arguments that must be made, rather than whether an assessment should be done or not. In the common case where a change is made to a non-safety module, the only statement necessary could be simply be a release note that says "this change does not affect safety critical modules."

In other words, the notion of "significant" should not be subjective. Rather, the process for re-assessment should be recast to permit submitters freedom to architect to minimize their safety assessment burden, while avoiding a loophole that evades re-assessment of safety critical functions when they have in fact been modified.

**Topic #4: Requiring Reassessment Of Fall-Back Triggering Changes To Fall-Back System**

**Recommendation: Reassessment should be required for any modification that can potentially affect the triggering of a fall-back strategy, including any modification to a module which self-reports failure as a basis for triggering the fall-back strategy.**

*Discussion (see policy pages 17, 30):*

As is stated in the policy, it is important to trigger a safety re-assessment when the fall-back strategy or its implementation has been modified by *any* change. (Note that the policy document appropriately omits the "significant" change threshold limitation here). However, this leaves open the problem of actually detecting that the HAV system is having a problem so as to trigger the fall-back strategy. If an autonomy system self-reports problems to trigger the fall-back strategy, then a change in the normal autonomy function could have a bug that that cripples the fall-back trigger.

For example, a self-diagnosis bug can cause missed problems; such a self-reporting diagnosis failure could render the vehicle unsafe. As another example, relying on an autonomy "heart beat" to diagnose failure can suffer from a bug in which the heart beat is running even though other significant autonomy features within the autonomy system have failed.

The policy should expand the scope of this section to include both changes in the fall-back strategy and implementation *as well as* any change that potentially affects the ability to trigger the activation of that fall-back strategy. This would include any change to primary autonomy unless it can be argued using a realistic and appropriate fault model (one more thorough than simply a "fail crash" assumption) that the fall-back strategy will activate regardless of any fault in the primary autonomy system, including self-diagnosis faults. Thus, activating a fall-back strategy by some means independent of primary autonomy self-diagnosis is likely to be a better architectural choice.

It should be noted that self-diagnosis is generally unsuitable for detecting life-critical failures because of the likelihood of an internal fault that compromises both functionality and the self-diagnosis capability. For example, see: *Hammet, Design by extrapolation: an evaluation of fault-tolerant avionics, IEEE Aerospace and Electronic Systems, 17(4), 2002, pp. 17-25.*

**Topic #5: Define A Methodology For Determining "Reasonable" Exceptional Scenarios**

**Recommendation: A methodology should be defined to provide guidance and a robust suite of examples for what exceptional operational scenarios could "reasonably" anticipated during system design and testing, accounting for the reality of what will happen over billions of hours of fleet operation.**

*Discussion (see policy page 30):*

The degree to which a crash or other mishap "reasonably could be anticipated" per policy wording is in the eye of the beholder, and is likely to cause many problems downstream if implemented without further elaboration. A significant problem is that a lay person interpretation of "reasonable" is often tied to personal experience, which is highly variable, and generally small in scope compared to all the things that will be experienced by a large fleet of vehicles. In other words, scenarios that a typical lay driver will consider likely to happen would be far too restrictive a standard to apply when considering what exceptional scenarios should be included in design and testing. On the other hand, "reasonable" could be something that happened once at any time in the history of driving (on the theory that if it could happen once, it could happen again, and therefore is foreseeable), and there might be a situation in which this is an unduly high standard. There are no doubt many other possible definitions.

An additional problem is that if designers take too restrictive a view of what scenarios "reasonably" could be anticipated, they could be forced into huge numbers of software updates as unusual events happen to the fleet over time. Each such update could require significant system validation and testing cost to assure safety due to the changed vehicle behavior. This would especially be true if the threshold of a "reasonably foreseeable" event were deemed by the Court system to be whether a mishap scenario had ever happened before to any single vehicle in the national HAV fleet.

This section would be much stronger, less subjective, and less subject to downstream surprises if it were tied to a numeric target or a proxy for a numeric target, even if such a target is simply a partial definition of what is reasonably foreseeable. For example, one could propose that reasonably foreseeable events include all events that are expected to happen to at least once every ten billion operational miles based on analysis and accumulated fleet experience. Such a target might rule in events that seem unlikely in everyday experience but happen in the real world. Assuming appropriate data support, foreseeable scenarios might include bridge collapse, tornado evasion, or operation after vehicle lightning strike.

By the same token, such a target might rule out events if objective data supports they are truly one-off events, if they actually do happen by chance to one vehicle somewhere in the fleet (e.g., potentially a vehicle hit by meteor strike might be deemed too unlikely to be included as a test case). Tying such an analysis to a "failure budget" that permits omitting extremely rare and expensive-to-deal-with scenarios *but does not omit frequent-enough scenarios* would help with being more rigorous about the claimed safety benefits of autonomous vehicles compared to human drivers.

It is arguably likely that system designers will under-estimate the cumulative safety risk caused by very rare but inevitable events, especially if they have not personally experienced a particular problem in the context of their career. As an illustration, consider that even a 100-year-old person has lived less than a million hours (876,600 hours), and so is unlikely to have seen a once-per-million-hour rare event while actually driving a vehicle manually. It might be difficult to believe that something that one hasn't seen

for oneself will actually happen. However, such a once-per-million-hour event will on average happen *numerous times per day* to vehicles within a deployed HAV fleet having millions of vehicles. Thus, a numerically base decision criterion for including or excluding exceptional scenarios will provide a more objective, defensible way to determine whether an event should be including in design, validation, and testing processes.

An alternative approach might be for DoT to build a dictionary of reasonably foreseeable events that is periodically updated. This could form a minimum set of situations that would have to be handled in terms of accepted practices of building such vehicles.

**Topic #6: Assure The Integrity Of Self-Reported Crash Data**

**Recommendation: The integrity of self-reported data relevant to safety and mishaps must be assured via independence between the data source and system components potentially responsible for the mishap or other safety incident.**

*Discussion (see policy pages 17-18):*

The current policy discussion centers on data recording for capturing and analyzing driving situations, which is surely important. However, a significant issue with data recording is the credibility of the data itself and drawing conclusions about ground truth based on interpreted data. If the data being recorded is the output of a computational process that might be defective, that data cannot be used to exonerate the computational process from having defects, since it is telling you what it "thinks" is happening rather than what is actually happening. As an example, if a radar says there is no object present, that simply means the radar did not register an object, and is not definitive proof that no object is really there. Similarly, if the autonomy system says the driver "disengaged" the autonomy system in a data recording, that simply means that the autonomy system "thought" it was being disengaged, and is neither definitive proof that the driver actually disengaged the system, nor proof that the software actually relinquished control of vehicle operation to the driver. There can always be a system design or operational defect that affects the ability of any particular system component to self-report its status and actions accurately unless it has been built with an appropriate redundancy strategy (see the Hammet 2002 reference mentioned previously).

The requirement should be expanded to include that a safety assessment should analyze the credibility of various data that has been recorded, and that credibility of data should be maximized to the extent practicable. As a simple example, recorded vehicle speed should not solely be self-reported by the autonomy software, but rather should also come directly from a speed sensor reporting capability that is independent of the primary autonomy. In the alternative, a safety argument should be made as to why speed reports are reliable even if arbitrarily bad faults occur in autonomy software. Such an argument could, for example, include an ability to cross-check self-reported autonomy speed data with other indirect sensor data that does not pass through the autonomy software. The limitations and validity arguments relevant to such data should be made available to crash investigators to prevent mislaying blame for a mishap based on misunderstanding the trustworthiness of various aspects of the data recordings.

**Topic #7: Diagnostics Should Include Non-Collision Failures And Component Aging Effects**

**Recommendation: Diagnostic coverage should go beyond post-crash coverage to ensure vehicle safety even if components fail due to normal failure, and dependability calculations should account for the inevitable elevated failure rates displayed by aged components.**

*Discussion (see policy page 25):*

Beyond crashes, components can be expected to fail in normal operation as well. The statement that "When problems are diagnosed, the HAV should be maintained in a minimal risk condition until properly serviced" is good, but misses two essential points. First, this statement should apply to the entire life of the vehicle, not just post-crash behavior as implied by the section title (i.e., self-diagnosis must encompass component failures due to non-crash causes). Second, the statement should make it more explicit that diagnostics should continually provide extremely high coverage of all safety-relevant hardware and software component faults over the operational life of the vehicle to avoid missing latent faults that could compromise dependability.

As a simple example, if the fall-back strategy subsystem has failed, the vehicle diagnostic system must detect that failure and account for the fact that any safety argument based on a fall-back strategy being available has been invalidated until the vehicle has been repaired. That failure detection must account for normal in-service failures, not just post-crash failures.

Additionally, the topic of component lifetime limits was discussed at length in the government-sponsored Automated Highway Systems program in the 1990s, but does not seem to be included in this policy. Eventually, probabilistic hardware reliability arguments made about equipment (e.g., that the fall-back computer will actually work for a long enough time to transition the vehicle to a safe state when it is called upon) become invalid due to end-of-life wear-out and elevated electronic failure rates (typically referred to as the reliability bathtub curve). A safety assessment should include an analysis of viable component lifetime, and a strategy for ensuring safety when failure rates exceed permissible levels. This does not mean waiting for a component to fail, but rather recognizing that the probability of a component failure is too high to assure safety when a vehicle's electronics get too old. Responses to excessive component age might include disabling autonomy or reducing functionality pending replacement of aged-out components.

One way to resolve this concern would be to designate an expiration date for autonomy functions on a par with the normal expected operational lifetime of the vehicle, and ensure that enough redundancy is built into the vehicle for reliability numbers to work out satisfactorily over that lifetime. Alternately, some autonomy components that do not age well might have to be replaced periodically as part of normal scheduled maintenance.

**Topic #8: Uniformity In Traffic Rule Exceptions**

**Recommendation: Acceptable parameters for traffic rule exceptions should be made as uniform as possible across the HAV fleet.**

*Discussion (see policy pages 25-26):*

Dealing with exceptional situations in traffic will be crucial to the ultimate success of HAVs, as is recognized by section 10 of the proposed policy. The policy should go further and recommend that the industry collaborate on creating a more thoroughly specified set of traffic rules that take into account the numerous foreseeable exceptions to rules that many drivers encounter every day in the real world, even if they are rare. (See also Topic #5 above.)

In other words, HAVs present the opportunity to have better uniformity and transparency in what would constitute reasonable and safe behavior in such exceptional but everyday situations. At the least, a taxonomy of such special situations should be developed to establish a baseline set of requirements. (The list of normal driving found on pages 28-29 might provide a starting point, and as noted in endnote 37, starting down the path to flesh out a set of scenarios was the intention of the referenced work.) Such an approach would also give vehicle-makers a better ability to defend the actions of a vehicle that behaved according to norms for a particular situation. It will also enable NHTSA to better judge whether the behavior of a particular vehicle is acceptable, and whether testing of exceptional situation handling is complete enough.

It is likely to be challenging to fully define a set of likely exceptional driving situations and also reconcile such rules against human driver behaviors and subsequent behavior adaptations. However, such challenges cannot be avoided, and will have to be addressed as society comes to terms with mixed human and automated vehicle traffic flow.

Put another way, intentionally designed deviations from traffic rules should be authorized in a transparent and methodical way rather than being left at the discretion of HAV design and validation teams. Such an approach should ultimately benefit HAV designers by mitigating the risk of a recall due to a behavior they thought was acceptable that is later disallowed by the government.

**Topic #9: Driver Takeover Strategies Should Have Safety Assurance**

**Recommendation: The capability for a system to fall-back to manual driver control should be the subject of safety assessment, even for SAE Level 2 vehicles.**

*Discussion (see policy page 34):*

As is recognized by the policy and its adoption of SAE levels 1-5, autonomy is not a black-or-white proposition. If a safety critical aspect of vehicle operation is fully controlled by a computer (e.g., throttle-by-wire, brake-by-wire) then an argument that the human can take back control necessarily assumes that the software will actually cede control back to the human. (A purely electro-mechanical takeover mechanism is a possibility, but is often missing, inappropriate, or might be subverted by software defects.)

In Table 1, SAE Level 2 implicitly has a fall-back strategy, which is that the driver asserts control. It is critically important that ceding control to the driver be dependable at any autonomy level where the driver is permitted to demand control or is expected to exert control. It is entirely foreseeable that a software defect in the autonomy could prevent reversion of control to the driver if the system is not designed properly with this safety critical requirement in mind. Thus, F.3 should be “Partially” in this table.

Respectfully submitted,

Philip Koopman, Ph.D.

Carnegie Mellon University & Edge Case Research, LLC