# Truck Automation: Testing and Trusting the Virtual Driver

**Steven Underwood, Daniel Bartz, Alex Kade and Mark Crawford**

**Abstract** This chapter addresses the testing and evaluation of the virtual truck driver. While the primary focus of the discussion is on verification and validation in model-based systems engineering it also touches upon testing for certification, establishing regulations, public investment, and research and development. A reference architecture for automated driving coordinates designs at the vehicle and system levels for increased interoperability among components and improved efficiency. A model-based systems engineering approach exploits automated vehicle systems domain models as a primary means of information exchange to help manage the complexity and provide analytical support for efficient architecting, design, verification, and validation. These models support the testing and evaluation process for functional safety design and certification. Finally, demonstration pilots, operational testing, and natural use testing, combined with system design artifacts, are critical to public and regulatory acceptance of the virtual driver. Although safety must be assured, the primary challenge is how to make such assurances without relying on a human driver and vouching for the virtual driver under all allowable driving situations and conditions. This chapter provides some ideas on how all of this might come together and help bring fully automated vehicles to the market.

S. Underwood (✉)
Connected Vehicle Proving Center (CVPC), University of Michigan-Dearborn,
4901 Evergreen Road, Dearborn, MI 48128-1491, USA
e-mail: underw@umich.edu

D. Bartz
SAE Reference Architecture and Interfaces (RAI) Task Force, San Francisco, CA, USA
e-mail: danielbartz@gmail.com; danielbartz@autonomation.net

A. Kade
Ground Vehicle Robotics, US Army TARDEC, 6501 East 11 Mile Rd.,
AMSRD-TAR-R/264, Bldg.200C Rm. 1130C, Warren, MI 48397-5000, USA
e-mail: alex.kade.civ@mail.mil

M. Crawford
Research and Advanced Engineering, Ford Motor Company, Building 2,
20000 Rotunda Drive, Dearborn, MI 48124, USA
e-mail: markcraw@umich.edu



G. Meyer and S. Beiker (eds.), *Road Vehicle Automation 3*,
Lecture Notes in Mobility, DOI 10.1007/978-3-319-40503-2_8

**Keywords** Automotive · Trucking · Trucks · Fleets · Testing · Evaluation · Automation · Architecture · Verification · Validation · SysML · Safety · Army · Driving system · Simulation · Systems engineering · Functional requirements · Model-based · Reference architecture · Interfaces · Certification · Standards · Vehicle · Pilots · Operational testing

## 1 Introduction

Perhaps the easiest way to envision the automated truck driving system is to imagine what it takes to be an excellent truck driver. Among other things the excellent driver should be in good health and be perceptive, responsive and adept at maneuvering the truck and trailer in its immediate environment and in accordance with the rules of the road. Their senses are attuned to the state of the truck. The excellent driver knows safe stopping and following distances, and turning radii with and without the trailer under different conditions and loads. They keep the vehicle in good working order, up to code and to the extent possible, out of harm's way. This driver is also skilled at avoiding or steering-clear of road hazards including pedestrians, objects and vehicle events that could potentially cause crashes. And when things go wrong the driver is able to bring the vehicle to safely to a stop.

While this discussion about the excellent truck driver is not exhaustive it provides some insight into what is expected of the automated driving system and how it should behave in order to keep the passengers and cargo safe. It is much more than a collision warning or crash avoidance system; it drives the vehicle within selected bounds from beginning to end. For those in transportation safety it might suggest adding a new column for the virtual driver and a new row for near crash behaviors to the Haddon Matrix addressing pre through post-crash dynamics [1].

However, while Haddon focuses on the crash event, automated driving systems will address all aspects of human driving as well as new capabilities like short gap platooning and multi-vehicle coordination. The virtual driver has the potential to increase safety directly through more attentive perception and responsive handling of events, and indirectly by navigating at safe speeds, distances, and gaps that human drivers often neglect. The virtual truck driver may be able to follow a lead vehicle with a short time gap with greater safely and reliability than a human driver. These safe and extended driver behaviors are system level targets for testing, approval, and in many cases certification for automated trucking. The goal is to perform these behaviors more comfortably and safely time-after-time over years and across miles with fewer errors and with better performance and reliability than even excellent human drivers.

Testing and evaluation will help ensure a proper design the excellent virtual driver and build public trust as well as legislative and regulatory backing. System testing will also be necessary to assure consumer acceptance and responsible manufacturing and maintenance of self-driving trucks. Over the long term testing will help to improve design, grow trust, and future certification of automated trucks.

## 2 Autonomous Mobility Appliqué System

United States Army Tank Automotive Research, Development and Engineering Center (TARDEC) is taking an incremental approach to automation by integrating mature sensors and control systems into U.S. Army and Marine Corps tactical vehicles to assist drivers and enable future autonomous operation. This project and related architecture and test activities will be used to illustrate some of the test and evaluation themes addressed in the chapter.

The Autonomous Mobility Appliqué System (AMAS) will reduce the dangers of driving in combat by providing soldiers with Active Safety and Warning systems while at the same time providing as platform for incremental adoption of automated vehicle systems. The first phases of AMAS deployment will take the base platform and add semi-autonomous convoy control, relieving the soldier of the driving task, reducing fatigue, eliminating rear end collisions. Additionally these technologies have been proven to enhance operator situational awareness, to enable a more effective response to hostile situations. Future increments of AMAS will provide autonomously navigation through urban and rural environments and negotiation of pedestrians, intersections, and oncoming traffic. The TARDEC plans follow the following sequence of increasing functionality: (1) driver assist and active safety, (2) leader/follower behavior, and (3) full automation.

The full AMAS system consists of an integrated By-Wire/Active Safety Kit and an incremental Autonomy Kit. AMAS uses a combination of automotive and specialized sensing and localization systems to balance system performance and affordability. The AMAS system has been demonstrated on six different families of tactical trucks.

## 3 Engineering the Automated Driving System

The systems engineering process is the starting point for conceptualizing, designing and testing automated driving. These days the research and engineering communities are transitioning from low levels of vehicle automation to higher levels while, at the same time, leading edge automated vehicle systems are transitioning from research to product development. Trucks fleets, both commercial and military, are likely to be early adopters of automated driving systems, driven by fuel costs, availability of drivers and an increasing emphasis on safety. The trucking market has been early adopter of key precursors to automated driving including systems like Roll/Yaw Electronic Stability Control, supplemental Electric Power Steering, Lane Departure Warning, Automatic Emergency Braking and Adaptive Cruise Control.

While research is often less structured in the early stages it becomes more structured in the later testing phases. Truck research is addressing higher levels of automation in systems like Traffic Jam Assist, Automatic Trailer Backing Assist, Freeway Pilot, and automated off-highway hauling and queue movement that are

likely to be introduced the next 5 years. Engineers engaged in new product development follow a more highly structured systems engineering process with three phases: concept, development, and approval. Furthermore, developers will need to prove to the regulatory agencies (e.g., NHTSA, FMCSA, FHWA, Army Test and Evaluation Command, etc.) that these systems are safe and reliable before they are deployed.

The systems engineering process is presented in Fig. 1 along with methods for testing and evaluation depicted in red type below the associated activity box. In the concept phase system requirements guide the development of the initial proof of concept that is used to create a set of system specifications. Modeling and simulation are tools for testing the concepts for the hierarchy of systems, subsystems, software modules, and components.

It is beneficial to know up front how the system will be approved. One of the core challenges today in conceptualizing, designing, and testing the virtual driver is the lack of knowledge as to how the system will be approved and/or certified if that is required. Looking ahead to the future of approving automated vehicles for on road usage it is advantageous to prepare a plan for coordinating the system requirements and architectures with the approval processes and scope the test cases to take into account plans for later validation and verification.

Functional and non-functional requirements guide all phases of the process. The circles in the Fig. 1 indicate the iterative nature of the process. The concept phase generally leads off with an overall definition that includes a hazard analysis and risk assessment and then moves into development of a product and approval of a validated system. A formal validation and verification plan may include among other functional and nonfunctional requirements for truck automation:

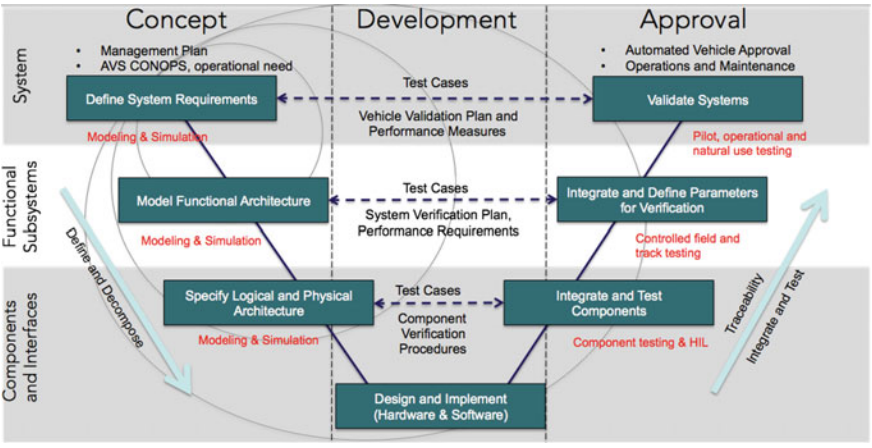


**Fig. 1** Testing and evaluation in the model-based systems engineering process

- Functional correctness: Does the system deliver the specified functions (e.g., localization, navigation, leader/follower, obstacle avoidance, conforms to rules of the road, passenger riding comfort, vehicle safing, etc.) and maneuvers (e.g., following, passing, stopping, etc.,) in target environments including weather, road geometry, traffic signals and signs, road conditions, lighting, traffic, etc. (i.e., functionality, maneuverability)
- Fault management: What happens when a fault is detected and how is safety and dependability managed in that context? Does automated navigation require a real-time operating system and alternative path planning in cases of a severe fault? (i.e., safety, dependability, reliability, redundancy, etc.)
- Safety and dependability: Is the system sufficiently available, reliable, and maintainable to assure safety and avoid catastrophic consequences? How does the system manage faults and near to complete system shut down? (i.e., availability, maintainability, reliability, safety, etc.).
- Performability: Assuming there is no driver to take over, will the automated vehicle systems perform robustly under constraints like the presence of multiple faults? What is the vehicle safing strategy?
- Acceptable cost: Does the system address all the other requirements at a reasonable and acceptable cost in the context of expected production volumes and market demand, penetration, and competition? Does the system support well-defined interfaces that will promote component level competition on design and production? (i.e., extensibility, interoperability, etc.)

Reliability engineering for automotive virtual driver is challenging because of consumer cost sensitivity and most automotive equipment is used until the end of its life. Common practices such as triplex redundancy of critical components may not be affordable in automobiles. In most safety-critical tasks, preventive maintenance schedules call for replacing electronics before the end of their design life. In the automotive environment, many components are never replaced unless they fail. Furthermore, at highway speeds it would help to have cost effective sensors that could sense up to 300 m and comply with functional safety and all target environments. Many challenges still exist.

Once the requirements have been laid out the overall performance and safety of the product can be established through the verification and validation stages of the development process.

## 4 Validation and Verification

The purpose of most product testing is to ensure that the as-built systems and sub-systems meet the design requirements and that the product meets the requirements of the end customer. The definitions we are using for verification and validation originate from the Defense Modeling and Simulation Organization (DMSO)

where verification is defined as "the process of determining if the model implementation and its associated data accurately represent the developer's conceptual description and specification." This requires some understanding the functional breakdown of the product. Whereas, validation is defined as "the process of determining the degree to which a model and its associated data provide an accurate representation of the real world from the perspective of the models intended use." In more common terms the main purpose of V&V is to answer key questions: (1) "are we building the right system?" (i.e., validation) and (2) "are we building the system right?", (i.e., verification).

Formal validation addresses whether the software or system does what the user really requires. Does it conform to the "customer" needs and the concept of operations?

Formal verification involves the specification of requirements and the design of a model or representation of the systems in a formal specification language that is semantically complete and allows for rigorous analysis. With the requirements expressed in a formal language the analysis can take on axiomatic or semantic forms. The axiomatic approach involves analytical reduction and mathematical analysis of the ability of the systems to achieve the specifications. The semantic approach involves model checking and uses exhaustive search through all possible program executions while looking for behaviors inconsistent with formally stated requirements. Both approaches are challenging when addressing automated driving in the context of functional safety.

The final evaluation and approval of these automated systems will address intended use of the automated system, use exceeding system limits, and use involving system faults. The evaluation depends on both the virtual driver's identification of critical situations and its ability to execute appropriate actions though a variety of maneuvers under a range of environmental conditions. The sheer magnitude of possible harsh environmental conditions, and possible critical situations poses additional serious challenges for design and acceptance of trustworthy virtual driving systems. An extremely improbable event, one that occurs once in 10 billion hour, would cause one failure in approximately 70 years in the US commercial air fleet. That same probability would cause a failure at least once a week in the US automotive fleet, due to the much higher number of vehicles. Even if the risk to a passenger might be the same in both cases, the public perception of risk could be much higher for road vehicles.

## 5 Trusting Automated Driving

Trust has a variety of meanings in systems engineering. In this context trust will assume a conventional meaning that is often used in trusted computing where the computer will behave consistently, in expected ways, and those behaviors will be

enforced by the design of computer hardware and software. This is consistent with everyday usage where trust is learned from logical assertions and positive experiences over time. The problem for testing automated driving is to provide logic evidence of system performance, safety, and reliability over time under all reasonable conditions.

Compounding the problem of testing trust is that automated driving systems require advanced perception, navigation, and control systems that are distributed across many hardware and software components. Most automated driving systems are built with adaptive, non-deterministic "intelligent" algorithms to address a broad set of environmental conditions.

Complex software code often manifests isolated faults under rare scenarios that are difficult to find and expensive to produce through traditional V&V approaches described above. If the V&V strategy is to field test on roads then hundreds of millions of test-miles are unlikely to detect low-rate systematic defects in software that operates an entire fleet of vehicles. Furthermore, certain events pose challenges to the virtual driver as well as the human driver including abrupt maneuvers at high speeds. Control systems that are generally optimized for smooth performance at cruise may not work for abrupt maneuvers in emergency situations. Furthermore, special systems and controllers need to be designed and adopted to cope with flat tires or loss of power for braking or steering. Moreover, engine and transmission dynamics are difficult to model at slow speeds while icy roads and other low friction surfaces always difficult to handle.

More complex systems provide addition challenges, as traditional V&V methods are difficult to scale with increases in system complexity, environmental diversity, and breadth of usage scenarios. Increasing system complexity demands more elaborate testing and additional expense to guarantee the safety and reliability of these systems of systems. The complexity of the software itself is a major driver of system complexity, and which according to Wagner and Koopman [2], exhibits:

- Millions of lines of code (planning and control of a driverless vehicle has considerably more operations than throttle control)
- "Cyclomatic" complexity needed to implement driving behaviors,
- Multidimensional-dimensional interfaces to transmit rich perceptual data
- New algorithms involving real-time control, machine learning, and adaptation.

This suggests that analytical and simulation test scenarios, including maneuver features and environmental conditions, can be rated for their pervasiveness, abruptness, delta speed, and presence of challenges that will not only provide insight into the specific behavior being tested but possibly suggest outcomes when conditions are changed. For example, while there will be exceptions, successful performance of a maneuver at a high speed could suggest that the same maneuver could be performed successfully at reduced speeds. An approach work considering is to thoughtfully "push the envelope" in the simulated environment to enhance our understanding of performance limits.

## 6 Model-Based Systems Engineering

The design and evaluation of automated driving require advances in systems engineering to manage the system complexity and improve traceability of changes and impacts throughout the process. Modeling and simulation techniques are relevant to testing the performance of automated vehicles cooperating in road networks with traffic control, vehicle-to-vehicle coordination as in CACC, convoys, and platoons, individual automated vehicles on various road environments and responding to dynamic or stationery obstacles, automated vehicle components, subsystems, and software modules including hardware in the loop, networks of wire line and wireless channels including databus traffic at interfaces and V2V and V2I communication, and human behavior in the vehicles to the extent it is relevant in the higher levels of automation. Furthermore, these forms of simulation can be creatively integrated to address specific issues. In the context of vehicle automation simulation is a model-based approach for helping address a broad range of automated scenarios or identification of specific "worst case" scenarios. It can also be used for more in-depth study of specific areas, as well as testing new features prior to implementation. New systems dynamics models with feedback loops can help the Tester address complex systems relationships and feedback mechanisms. The Systems Modeling Language (SysML) is a general-purpose modeling language that supports the model-based approach including the specification, analysis, design, verification and validation of automated vehicle systems.

Model checking is a key feature in trusting automated real-time control systems [3]. Wagner and Koopman [2] predict that automated driving will rely heavily on inductive inference and complex software to operate safely and that traditional software safety techniques are not up to the task of analyzing and mitigating the risks they pose. Wagner argues software testing should focus more on negative test results that motivate ongoing, iterative software improvement [2]. The model-based approach opens the door for more creative and automated generation and testing of scenarios. An important rule for software engineering is that a defined output should be provided for any possible input. So instead of testing how the architecture handles normal, "clean" inputs, the model-based approach can test abnormal, "noisy" inputs crafted to stress the system and identify potential vulnerabilities including software bugs, flawed architecture design, communication failures, environmental conditions that exceed the systems design parameters, inconsistent internal states, and gaps in system testing.

TARDEC has used this approach to test their automated vehicle systems by feeding unexpected inputs into the sub-system to identify abnormal behaviors. This reveals problems that often go undetected using other forms of testing. Examples might include safety invariants such as speed limits or not moving while the emergency stop is engaged. The model can look at violations of these and other tests and flag whether the vehicle exceeds the speed limit or violates other safety rules or if the subsystem experiences faults the lead to system crashes. In general the Tester can develop a test specification based on the architecture including an

interface definition and safety invariance that will be tested based on existing safety requirements. This information is used to generate a set of test cases during the test the system monitors output at run time for violations of the safety invariance.

## 7 Automated Driving Reference Architecture

The SAE Reference Architecture and Interfaces (RAI) Task Force is using Systems Modeling Language (SysML) general purpose modeling language to support specification, analysis design, verification and validation of the Automated Driver systems of systems. The SAE Reference Architecture uses nominal requirements, desirable properties, behavioral diagrams, threshold values for metrics, and structured diagrams to generate a SysML model that will aid in the analysis and requirement traceability required for verification and validation. This approach is being considered for model based identification and evaluation of real options to support strategic test planning. Model-based testing (MBT) relies on models of a system under test and/or models of its environment to derive test cases for the system [4].

A Reference Architecture helps facilitate a Model-Based Systems Engineering process in several ways: It provides a common language for systems design. It also provides best practices and design patterns that can aide in the development of safety and interoperability requirements for the virtual driver, particularly in early stage programs where many requirements are not well defined. It is a valuable tool for the systems engineering tool chest.

The SAE taskforce is continuing work started under the Interoperability Profile (IOP) activity initiated at U.S. Army Tank Automotive Research, Development and Engineering Center (TARDEC), Warren, Michigan. This taskforce is chartered to design high-level functional reference architecture for automated wheeled ground vehicles (e.g., military and civilian, trucks and passenger vehicles) covering use cases present in SAE Automation Levels (SAE J3016) 3 through 5 [5].

The RAI taskforce will identify possible standard work products and provide recommendations for extending existing vehicle messages sets (e.g. SAE AS-4 (JAUS), SAE J1939, SAE J2945, etc.) to encourage interoperability and reuse of automated vehicle of subsystems and components. Elements of existing standards like AUTomotive Open System ARchitecture (AUTOSAR), real-time operating system OSEK/VDX, Robot Operating System (ROS), LIN, CAN, FlexRay, and Ethernet are being leveraged to the extent that they applicable to fully automated vehicles.

Systems based on the reference architecture will need to address the hard-real-time requirements of vehicle control while intelligent monitoring of the driving environment on public and other drivable roads. This includes behaviors such as highway driving, obstacle avoidance, leader-follower, platooning, all the way up to full automation (i.e., SAE Level 5). The task force is following a process that includes the following: (1) review and mine patterns from state-of-the-art

system designs, (2) analyze and decompose requirements, and (3) model and integrate a scalable, flexible, functional reference architecture the supports these requirements. The first phase will be documented in a whitepaper that describes the working reference architecture and its requirements, modules, interfaces, and messages. While the first phase focuses on prior art the second phases focuses on capturing specific requirements, such as those described in the introduction, and derived requirements for elements such as spatial coordinate frames, real-time control system hierarchies, data management and high-fidelity maps, communications analysis, trust and reliability, and possible performance parameters. This background and analysis is feeding into the third activity, the modeling of a scalable functional architecture using SysML. It is intended that this activity will continue to evolve as the technology and requirements evolve and as new best practices are adopted.

Figure 2 shows the functional layout of the working draft of the reference architecture supporting the entire dynamic driving task. RAI's current focus is on the development of SysML models for Level 4 for vehicles including cars and trucks. Figure 2 covers functional blocks of the architecture such as sensing, perception, navigation, active safety and driver assist, and vehicle controls. The details

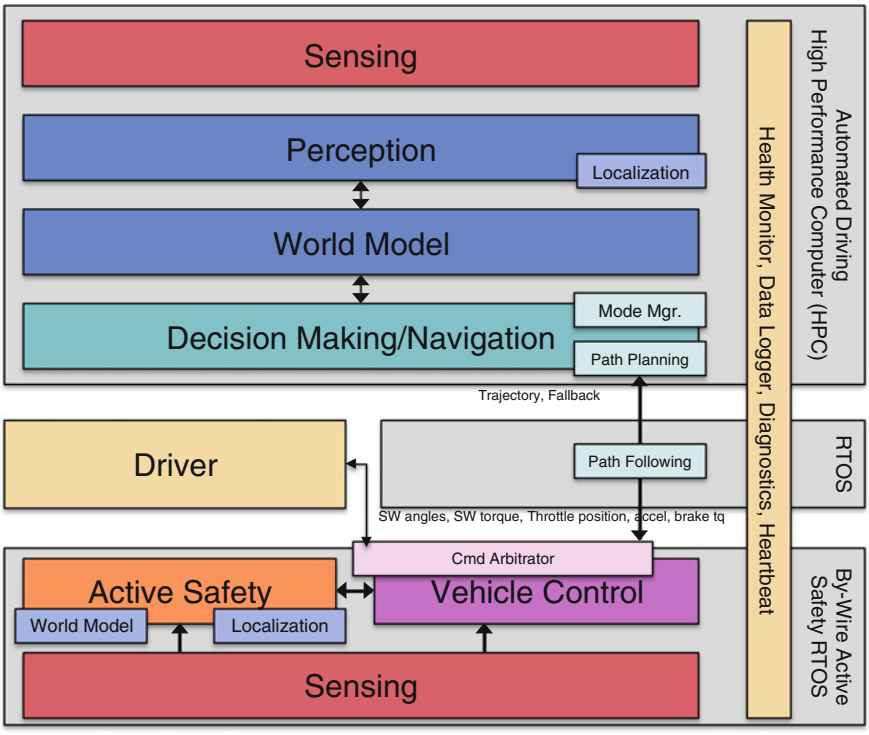


**Fig. 2** On-road automated vehicle reference architecture scope

of diagram were selected due to their wide use across the published literature or as lessons learned from actual implementations. This diagram is provided more for discussion than for definition. Today's Active Safety and Driver Assist (orange in Fig. 2) provide not only provide a baseline for automated functionality they may also evolve independently from higher automation systems, and provide assistance to human and proving a secondary check on virtual drivers. It is assumed that in a physical implementation critical modules will have redundancies of some form, these have been omitted from Figs. 2 and 3 to reduce their complexity. The colors in Fig. 2 are coordinated with the layered view of the reference architecture in Fig. 3.

The ORAV reference architecture will provide a structural framework and common language that will assist in consistent modular implementation and possible standard interfaces. The concept is similar to how AUTOSAR provides Application Interfaces that can aide verification, system interoperability, and module reuse. Similarly solutions emerging from RAI are designed to support both module interoperability and subsystem functionality.

The critical analysis of the performance requirements of key automated vehicle functionality will drive certain design elements of the interfaces between various software and hardware modules. A particular emphasis is being placed on the interface between the virtual driver and the vehicles by-wire control system. In order to insure that these interfaces will support robust real world systems RAI has found it necessary to analyze the critical elements of vehicle safing behavior, motion controls, message latency, extensibilty/expandability, cybersecurity, fault detection and tolerance, and safety critical data bus traffic.

The ORAV reference architecture is a work in progress. While at this stage there is no plan to produce a standard RAI is intended to be an authoritative source of information that captures best practices that can guide and possibly constrain

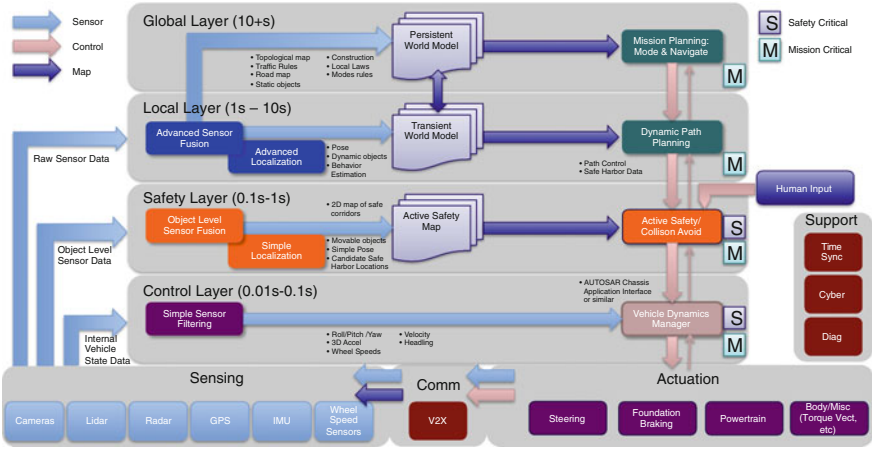


**Fig. 3** On-road automated vehicle architecture layers

instantiations of solution architectures (architectures of systems being developed). An useful reference architecture will state its purpose and offer guidelines and logic for how requirements will be achieved through a description of reusable patterns of activity, functionality, information/data flows, and scenarios of sequenced responses to time events. It will provide guidance and structure at the logical level of (software) modules and messages that flow between the modules while taking into account the general use of the product and the related requirements, functions, and activities.

An important early focus is to evaluate the impact that system safety has on the design. The central approach to functional safety is to design the automated driving system so as to minimize unacceptable risk of physical injury or of damage to the health of people either directly or indirectly [6]. Timely driver takeover cannot be guaranteed in levels 4 and 5 automated vehicle systems. This means that these systems must be able to continue to operate safely, even under fault conditions, until they can be brought to a safe state. This requires identifying hazards and the required safety functions, assessing the risk reduction required by the safety function, and implementing appropriate risk mitigation techniques to ensure fault-tolerant performance in all conditions. Again, this is a challenging task.

The high degree of complexity required for vehicle automation combined with a high degree of safety-critical performance makes the verification a daunting task. The ORAV reference architecture leverages key techniques from NIST's RCS (Real-time Control System) framework [7, 8]. The RCS framework was also used to develop the 4D-RCS architecture for automated systems used by the DoD. Classical systems engineering manages verification by decomposing complex systems into subsystems, often along functional lines. The RCS framework seeks to achieve the balance between intelligent system performance and ease of verification by decomposing systems both by functionality and by time domain. A control system is decomposed into multiple layers based on a given functions temporal/spatial span-of-control requirements [7]. Each layer is allowed to plan forward and look behind a finite number of time steps (NIST recommends 10). Higher layer have time steps an order of magnitude longer than the layer below it. For example a layer may work with 20 time steps ($\pm 10$) of 100 ms each. The layer below it deals with 20 time steps of 10 ms. In this way continuous control is achieved while the complexity and state-space of each layer is bounded. This makes verification complex systems more tractable than they would if they had been decomposed along only functional lines.

This is combined with the functional safety concept of "mixed criticality". In this approach final safety authority is consolidated in key highly verified subsystems. This allows other subsystems to be certified at lower criticality levels while bounding system-level risk. This type of practice is used in air systems, where often a high-performance flight controller is shadowed by a simpler, safety-critical flight controller(s). The secondary flight controller(s) take control when it is detected that the primary flight controller has issued faulty, inappropriate or no commands at all. The secondary controller must maintain control until the primary system is recovered or the aircraft brought to a safe state (landed). While the complex, the

primary flight controller is capable of high degree of performance (fuel economy, smoothness, etc.), the simpler design of the secondary flight controller allows it to be verified.

Figure 3 shows the same system as Fig. 2 decomposed into layers based time horizon and safety criticality. The proposed RAI architecture is broken into four notional layers, starting from the bottom:

1. Control Layer: The lowest/fastest layer deals with actuator control and vehicle dynamics management based mostly on internal state sensors (e.g., acceleration, gyro, wheel-speed, etc.). This would include traditional features like ABS, and ESC. This layer would need to be fault tolerant and hard-real-time.
2. Safety Layer: This layer the deals with simple maneuvers based on simple, robust external sensor data (such as radar object tracks). In manual vehicle operation is provides traditional ADAS and collision avoidance functionality. Under automated control it acts as the safety critical backup to the Local Layer. This layer will execute the Safe-Harbor maneuver in the event of the failure of higher level automation. An Active Safety Map (e.g. ADASIS) could be used to aid operation and Safe-Harbor functionality. This layer would need to be fault tolerant and hard real-time.
3. Local Layer: This layers deals with automated driving within sensor range, including fusing high-level sensor data, tracking other vehicles, projecting their behavior, populating the Dynamic Map, and planning vehicle maneuvers. It may also provide vehicle safing artifacts to the safety layer (such as suggested vehicle safing path).
4. Global Layer: This layer deals with beyond-sensor horizon planning, perceiving the static elements of the world around the vehicle, maintaining the long term static map, and planning the driving route (mission planning)

A rough analogy can be made with the human nervous system [9]. The Safety and Local layers are similar to the dorsal/ventral model of human perception. The dorsal visual system is a high-speed, highly deterministic pathway deals with core collision avoidance and precise tracking. The dorsal system can controls a person's movement without conscious input. This is in contrast the slower, less deterministic ventral visual system that deals with recognition, and identification and is a core input to conscience decision-making. The Global Layer and Control Layer are similar to prospective memory/planning and the cerebellum respectively.

## 8 Standards and Certification Testing

Testing will be needed for certification to specific standards developed for automated driving systems and to provide evidence for meeting future regulatory requirements. This may include voluntary standards, industry verification and validation processes, and government assessment processes. In the automotive

industry this often involves laboratory testing of components combined with controlled tests on the test track.

In the United States, automotive safety certification is the responsibility of the OEM and their suppliers. Certification is based on compliance to engineering standards and recommended practices, as established by the organization (including SAE, IEEE, and TMC), and each vehicle/vehicle equipment manufacturer must test and certify that each motor vehicle and/or equipment item is in full compliance with the minimum performance requirements of all applicable Federal Motor Vehicle Safety Standards (FMVSS) set by NHTSA (e.g., Code of Federal Regulations, Title 49, Part 571). FMVSS are federal regulations specifying design, construction, performance, and durability requirements for motor vehicles and regulated automobile safety-related components, systems, and design features. Manufactures must confirm with the NHTSA that their products conform to the relevant standards through the process of self-certification. The NHTSA can inspect any product at any time to evaluate whether a vehicle or equipment item conforms to the performance requirements. For example, FMVSS 121 applies to trucks, buses and trailers equipped with air brake systems, but with some exceptions based on size, speed and weight, and FMVSS 105 applies to multi-purpose passenger vehicles, trucks and buses with a gross vehicle weight rating of 3.5 tons that are equipped with hydraulic or electric brake systems. The standards specify the test procedures that must be used for the purposes of self-certification, which include ASTM and SAE as well as procedures defined in the FMVSS document. The New Car Assessment Program (NCAP) also supports some voluntary third party testing and certification of automobile safety. Unlike the US, in Europe it is common practice for most testing to be done by third parties.

Automated driving is different from traditional active safety features because instead of aiding a driver, the system takes on the full driving task. Current standards, such as ISO-26262, assume that a human driver is in control and can mitigate faulty systems as long as these systems exhibit fail-safe behavior. Higher levels of automated driving must mitigate their own faults, maintaining fail-operational performance until the vehicle is brought to a safe state. At a conceptual level, tests for validation of automated driving Levels 4 and 5 is more analogous to human driver testing and may address, among other things:

- Basic maneuvering on surface streets, freeways, at intersections, in parking lots, etc.
- Maneuvering the vehicle to safety in case of a hazard or vehicle malfunction (i.e. minimal risk condition or safe state)
- Defensive driving and crash avoidance,
- Compliance with rules of the road, and
- The ability to recognize and handle complex, previously unseen scenarios.

Most of these fall under the category of performance testing with experimental controls providing an unbiased way of presenting the benefits or potential dangers of active safety systems on the test track. This approach requires the selection of

specific geometric road configurations and accompanying traffic scenarios based on performance requirements. It also requires specification of driver behavior and vehicle maneuvers as well as the positions of target vehicles and other obstacles in the scenario.

One could speculate that it might be sufficient for safety certification to prove the automated driving system can detect faults and bring itself into a safe state under all specified situational and fault conditions. In the previous section we mention the simpler safety-critical controller in the Safety Layer assuming this role and serving as a check on the Local Layer trajectories and Global Layer plans. Certification of the system could focus on the performance of the Safety Layer and the Vehicle Safing software.

In addition to performance and behavioral certification standards there are likely be industry standards dealing with structures, devices, and interfaces. These may include elements described in the reference architecture such as interfaces between the virtual driver and the vehicle sub-system including the vehicle control systems, sensors, and communication devices. Other examples include standards for sensor interfaces, 3D prior localization maps, diagnostics, and event data recording. Because of the number of potential use cases and the complexity of the software, the regulating agencies need holistic automated driver testing strategies/methods. These may include model-based approaches supported by simulation or hardware in the loop (HIL) testing, as well as "edge case" or "worst case" philosophy in the selection of test scenarios.

## 9 Demonstration Pilots and Operational Testing

While low mileage pilots can demonstrate the feasibility of automated driving systems in controlled environments, human supervised high mileage operational testing offers opportunities to encounter novel situations in the on-road environment, to make design modifications based on road experiences, as well as to validate and increase trust for vehicle safety in realistic environments over time. The pilot projects of automated driving will generally involve operational testing once the concept is in place and the selected automated system is well down the path of development. In other words, a proof of concept has been completed and the safety requirements are in place along with the verified design and functional Human Machine Interface (HMI). In most cases, the HMI will been verified through a concept simulation and driving situations will have been tested in a driving simulator. Test track evaluation using professional test drivers will also been completed. Tests for navigation include blind path tracking tests, perception-assisted path tracking test, and perception planning test in a broad range of environments and alternative use conditions. Operational tests usually include experimental designs with specific validation and verification goals that focus on error detection and ensuring compliance in typical operating environments. It facilitates consideration of environmental factors that influence system behavior and allows feature interaction.

More advanced approaches to operational testing of automated vehicles could include human and virtual driver learning and continuous system improvements throughout the test process.

TARDEC's Joint Capability Technology Demonstration (JCTD) demonstrated a 10 truck automated convoy with six different vehicle types. It tested these systems' ability to avoid simulated obstacles and pedestrians, and to mitigate accidents commonly caused by driver inattention or distraction. The evaluation also included user (driver and remote operator) interfaces. Cyber security is a serious issue for the DoD, and an extensive "red-teaming" of these systems has been performed. This included vulnerability testing with physical access (i.e. laptop plugged into vehicle systems) as well as high-power jamming of radios, lidar and automotive radar systems.

While much of the DoD operational testing in the United States is conducted on based or military test areas some of the testing for connected vehicle applications in road traffic will be conducted on the interstate highway system. At the time of writing the states that allow on-road testing includes California, Florida, Nevada, District of Columbia, and Michigan. The proximity of Michigan's I-69 corridor to TARDEC makes if a convenient site for on-road testing the vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication features of the AMAS concept.

A planned on-road test in 2016 will investigate 4 truck automated convoys along the Michigan I-69 freeway corridor between Port Huron and Flint where the vehicles will be traveling in close proximity to one another. This convoy will use V2V communications between the linked pair of trucks and is not dependent on V2V communications from non-convoy vehicles. The control system enables shorter distance and time gaps between vehicles in the convoy in a freeway environment with the potential to reduce fuel consumption and emissions, increase road throughput, and minimize driver fatigue associated with the long driving durations. The lead vehicle of this convoy will be manually driven. The trailing vehicles will follow the lead in sequence with distances of 20–30 m depending on the vehicle speed and breaking capabilities. While in this test, the drivers of the following vehicles will monitor the system. In actual use these drivers will have been situational awareness as they can focus their attention on the broader situational context. Drivers of the following vehicle must be in position to resume control with an increase lead-time if a takeover request from the system occurs. In this test the driver the vehicle should be capable of reacting to emergency braking maneuvers of their leading vehicle (Fig. 4).

While operational testing may be required to increase consumer and public trust for automated trucking it may not be as important an element in acceptance as durability and reliability testing. It is widely accepted in the automated vehicle field that these vehicles will need an abundance of time and miles for simple durability and reliability testing. It may be that the real value of operational testing is the ability to observe the vehicles behaviors under a wide variety of situations. These observations would be used in the development of better tests and more effective behaviors. The combined value of testing and learning over many miles and long durations of time is the identifications of important edge cases while increasing trust in these systems as improvements are made.

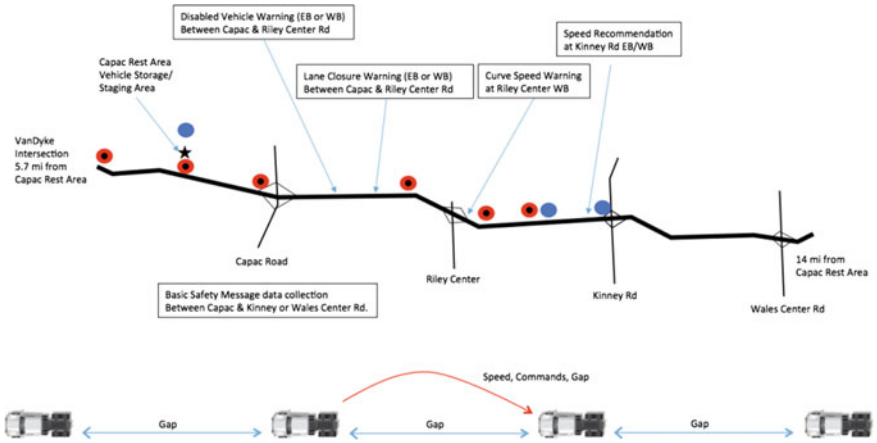


**Fig. 4** Operational test for Michigan truck caravan

## 10 Conclusion

One of the themes at the Automated Vehicle Symposium was the identification of new approaches for testing and evaluating in support of efficient engineering, certification, and consumer acceptance of automated driving systems. This chapter details and expands upon this theme from the perspective of the military truck panel in the breakout session on Truck Automation. Although demonstration and testing of platooning was discussed at length in other panels other of this session the primary focus on this panel was unmanned leader and follower behavior of trucks in military convoys. We address the testing and trusting of the virtual driver, or automated driving system, at the higher levels of automation where it is presumed a human driver is not available as a backup to the automated driving system.

While the design and operation of self driving trucks poses research and engineering challenges the parallel development of new methods for testing and certifying the virtual driver may be the most difficult obstacle in the path leading the way to public acceptance and trust of truck automation. Although current modeling, simulation, testing and evaluation methods have been effective for millions of commercial and military trucks in service, they are much less effective for the evaluation of learning and non-deterministic systems are being developed to support higher levels of autonomy. New approaches are needed to address the complexity and diversity of systems and operating environments that automated trucks will be used in.

The professional community is working on developing a methods and procedures for testing and evaluating the automated vehicles that draws on historical methods of modeling and empirical testing while investigating new adjustments and strategies for testing the performance and safety of the virtual driver. There is a need for standard and accepted test procedures. While many good tools are

available for modeling, testing, and simulation; however these can find problems but can't prove the goodness of an automated driving system. We conclude with several ideas that may help along out path to a new method for verification and validation of automated vehicles.

First, it is highly desirable to have a standard reference architecture that clearly defines and delineates the roles and functions of the virtual driver and the automated vehicle, as well as a standard interface between these entities. This would greatly facilitate the development of standards that could be used to validate and certify the higher levels of automated driving for use around human beings. In addition, this approach will enable continuous improvement of these systems, with OEMs, vendors, and regulators being able to focus on a common system, rather than diffuse efforts on multiple approaches. Furthermore, functional safety standards like ISO 26262 and its associated ASILs need to be extended and upgraded to take into account automated driving systems.

Second, system models must be developed to describe complete use cases and capture the function and structure of the system. These use cases can then be used to generate the proper test scenarios for V&V testing. The complexity of the driving environment makes objectively measuring risk difficult, and compounds the ever-increasing cost of redesign due to errors found in late developmental and operational testing. The diversity of the driving tasks makes it impractical to test under all conditions. Novel methods and approaches are needed to address this.

Third, high fidelity modeling and simulation must be heavily used to pre-test system behaviors, with specifically selected physical testing performed to validate the simulated responses. Forcing failures parallel simulations to expose issues. Better tools needed for massive (greater than terabyte) data collection, data mining and scenario recreation. Innovations in "worst cases" approaches to simulation and modeling are needed.

Finally, safe and correct operation of automated driving must be repeatedly demonstrated with real vehicles and broadly advertised to build trust and acceptance with the public. Extensive testing is required but not sufficient for dynamic real-time learning and adaptive automation systems.

## References

1. Haddon W Jr (1980) Advances in the epidemiology of injuries as a basis for public policy. Public Health Reports 95(5):411
2. Wagner M, Koopman P (2015) A philosophy for developing trust in self-driving cars. In: Meyer G, Beiker S (eds) Road Vehicle Automation 2, Springer International Publishing, pp 163–171
3. Goldman RP, Musliner DJ, Pelican MJ (2000) Using model checking to plan hard real-time controllers. AIPS Workshop on Model-Theoretic Approaches to Planning
4. Utting M, Pretschner A, Legeard B (2012) A taxonomy of model-based testing approaches. Softw Test Verif Reliab 22(5):297–312

5. SAE International, J3016—taxonomy and definitions for terms related to on-road autonomous vehicles. http://www.sae.org/works/documentHome.do?docID=J3016&inputPage=wIpSdOcDeTaIlS&comtID=TEVAVS
6. ISO 26262–3:2011(en) Road vehicles – Functional safety – Part 3: Concept phase, International Standardization Organization
7. Albus JS (2002) 4D/RCS – a reference model architecture for intelligent unmanned ground vehicles. In: AeroSense. International Society for Optics and Photonics, pp 303–310
8. Albus J, Huang HM, Lacaze A, Schneier M, Juberts M, Scott H, Murphy K (2002). 4D/RCS: A reference model architecture for unmanned vehicle systems version 2.0. National Institute of Standards and Technology (NIST). US Department of Commerce, Gaithersburg, Maryland 20899
9. Albus J (2008) Toward a computational theory of mind. J Mind Theory 1(1):1–38