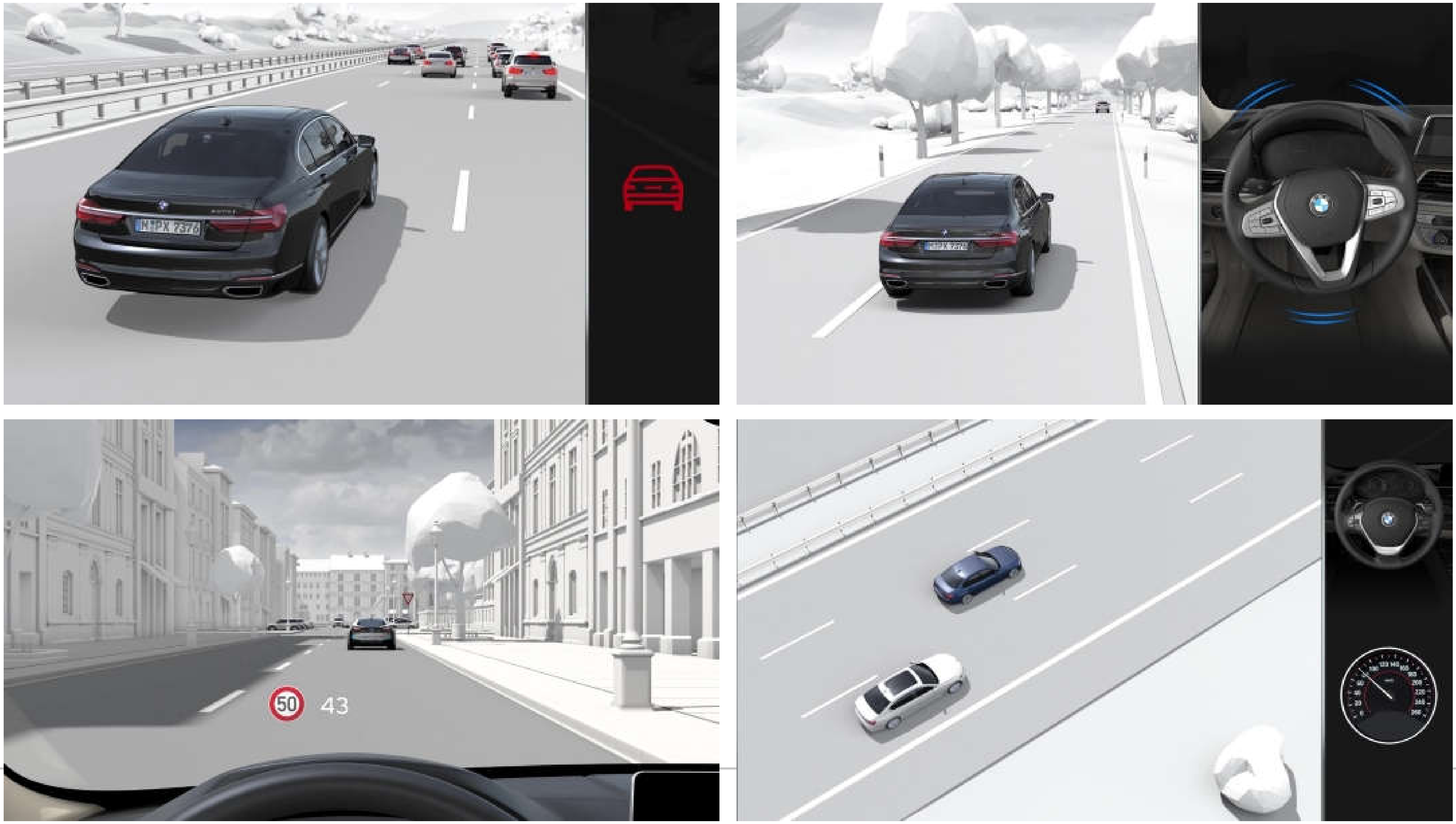
BMW GROUP

USE-CASES OF OPENSCENARIO.

THE NEED FOR A HIGH-LEVEL EXTENSION.

DR.-ING. MOHAMMAD BAHRAM
DR.-ING. LUDWIG FRIEDMANN
FEB 2019

USE-CASES.



EFECTIVENESS ANALYSIS OF DRIVING FUNCTION.

- Simulation systems like OpenPASS (https://openpass.eclipse.org/) are used to assess the effectiveness of driving functions during early pre-development.
- Simplified models of sensors, driving functions and physical models are employed in stochastic scenarios in order to generate aggregate results on the effectiveness of basic approaches and implementations in realistic traffic.
- This use case is characterized by the use of small, less complex scenarios, which can be edited by hand using a text editor. It employs descriptive stochastic scenario description elements with a focus on generated stochastic traffic to analyze the interaction between the ego vehicle and that randomized traffic.

Characteristics

- Minor usage of scripting, traffic represented by statistical models
- Limited scenario size and complexity
- Manual scenario creation in text editor

DRIVING SIMULATION WITH HUMAN DRIVER.

- Driving simulators are employed to analyze the interaction between human drivers and vehicle behavior, including driving assistance functions and autonomous driving.
- Scenarios are characterized by the need to model complex traffic interactions to the scripted human driver behavior. They are potentially large in size (e.g. complex inner-city traffic across many blocks), and need to be specified using flexible, reactive trigger conditions to enable the traffic to adjust to the actual human driver behavior.
- Support for the creation of complex scenarios using generative approaches, where scenarios are built up from sub-scenarios that are matched to certain map features and then instantiated as many times as they are present in a certain map, is essential.
- Scenarios are written in text editors by experienced scenario designers based on high-level specifications provided by developers and project/program managers of the relevant driving functions.

Characteristics

- Human driver, scripted scenarios and complex conditions
- Potentially large scenarios
- Manual scenario creation using graphical editors and scripting (text editor)

CLUSTER-BASED VALIDATION TESTING OF AUTONOMOUS DRIVING FUNCTION.

- Scenarios for validation testing of autonomous driving functions are potentially derived from many sources (regulatory agencies, in-house departments, crash testing agencies, research projects, etc.) and are likely to be managed and even generated by separate processes prior to their employment in validation testing.
- The scenarios are usually small in nature, with limited complexity and fixed trajectories for traffic participants. Portability and reproducibility across simulation tools are of paramount importance. Due to the expected large number of scenarios (including variations), efficient simulation of those scenarios in multi-job fashion on large clusters is required.
- Scenario generation is likely to use at least semi-automated if not fully automated processes, however reviewability of so generated scenarios is still likely to be required.

Characteristics

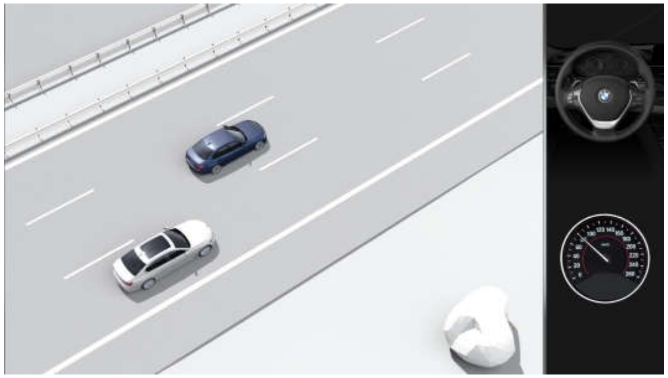
- Traffic modelled by fixed trajectories
- Varying scenario size
- Scenario creation by upstream software

TESTING DURING AUTONOMOUS DRIVING FUNCTION DEVELOPMENT.

- Lightweight support of ad-hoc test creation during development of autonomous function development by developers is centered around the creation of mostly simple scenarios that test specific behaviors of the function being developed.
- This should be supported by graphical editors to create and edit such scenarios visually, while still supporting simple parameter changes and adjustments through simple textual changes in the created scenario files.
- Easy comparability and mergeability of changes with line-based diffing tools commonly found in version control systems supports parallel development of functions and tests in tandem (i.e. in test-driven development approaches).

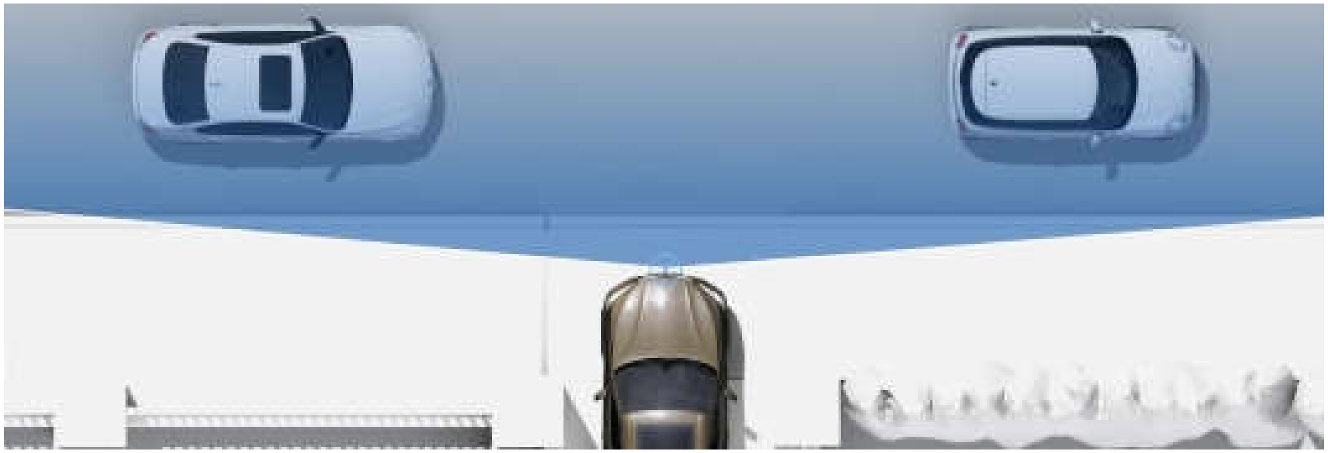
Characteristics

- Limited scenario size
- Varying scenario complexity
- Scenario creation by graphical / text editor



SCENARIOS AS CONSTRAINTS ON ACTUAL SCENARIO EXECUTION.

- While most scenario approaches treat a scenario description as something to be executed, another view of a scenario is as a constraint on actual scenario execution, to check whether a driven scenario effected through any means actually matches, within certain bounds, a scenario as specified in a scenario description. This approach can be used for example to analyze whether a set of executed scenarios has effected all major scenarios specified e.g. in legal requirements.
- To use this approach, the scenario description language must ideally be purely descriptive, without imperative and stateful aspects of scenario execution being interwoven with the scenario description itself.



SUMMARY.

1. **Effectiveness Analysis**
   - Minor usage of scripting, traffic represented by statistical models
   - Limited scenario size and complexity
   - Manual scenario creation in text editor

2. **Driving Simulation**
   - Human driver, scripted scenarios and complex conditions
   - Potentially large scenarios
   - Manual scenario creation using graphical editors and scripting (text editor)

3. **(Cluster-)Simulation / Test of AD Function**
   - Traffic modelled by fixed trajectories
   - Varying scenario size
   - Scenario creation by upstream software

4. **Development-Related Testing**
   - Limited scenario size
   - Varying scenario complexity
   - Scenario creation by graphical / text editor

CONCLUSION.

- Use-cases show very different characteristics and requirements for the scenario description language. The latter has to be formally standardized and be flexible enough to cope with the requirements.

- The workflow for the creation and review of scenarios must be scalable to meet the requirements of mass deployment and the use of scenarios of varying size (even very large ones). Feature-based scenario generation mechanisms can support developers with this difficult task.

- It has to be possible for users without programming knowledge to generate and edit scenarios. Therefore, both text editors and graphical editors should be able to be used for the scenario creation and review. The language used to describe the scenarios should be concise, unmistakable and intuitive. That requires the usage of a high-level description language.

- While a high-level description language solves many of the problems mentioned above, it must be possible to convert at least parts of the scenario description into a mathematically exact, machine-readable description. This low-level description language is required for the exact formulation of critical situations and tests. A reference implementation is required to enable the translation of high-level description elements into low-level description elements.