PEGASUS

PEGASUS METHOD

An Overview

Supported by:

Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag

Table of Content

1 Abstract

In the past years, many projects and companies have highlighted the public's interest in automated driving functions and have thereby focused on functional development. These show cases demonstrate a nearly possible series production of automated driving functions.

However, these demonstrations have shown only the functional view of automated driving and concentrate neither on the test nor on the verification and validation process for such automated driving systems. For advanced driving assistance systems, distance-based test approaches are currently used. Wachenfeld & Winner (Wachenfeld & Winner, 2015) estimate in a thought experiment the required test distance for verification of an automated driving function with the operation design domain (ODD) highway, where the required distance is approximately 6.22 billion kilometers to show that the automated driving vehicle is twice as good as a human driver. This demonstrates the disproportionate effort necessary to test automated driving functions through a distance-based approach.

Unfortunately, there are currently no state-of-the-art methods available, which can be directly applied in order to escape this dilemma of examining unrealistic billions of test kilometers. Therefore, new methods are necessary for efficient testing and for verifying and validating automated driving functions. Thus, research has to define a new general state of the art for test methods and test case selection, which can be used for a series release of these driving functions.

One possible approach, which is also used in other domains, such as software testing, etc., is a scenario-based approach for the testing, verification and validation of automated functions. This offers the advantage of a systematic and structured approach instead of a distance-based approach with random test cases. However, changing the approach also raises new (research) questions. Two examples are: *"What level of performance is expected of an automated driving system?"* and *"How can we verify that it achieves the desired performance consistently?"*

The research project PEGASUS (**P**roject for the **E**stablishment of **G**enerally **A**ccepted quality criteria, tools and methods as well as **S**cenarios and **S**ituations) on the release of highly-automated driving functions addresses such research questions using the example of a highway chauffeur ODD (operational design domain).

PEGASUS is promoted by the German Federal Ministry for Economic Affairs and Energy (BMWi). The project is split into four subprojects: *SP1: Scenario Analysis & Quality Measures, SP2: Implementation Process, SP 3: Testing, and SP 4: Reflection of Results and Embedding.*

The subproject 1 *Scenario Analysis & Quality Measures* addresses within the project the following questions and identifies the outlined results:

What is the human driving performance within the ODD (operational design domain)?

To answer the question, human driving behavior was analyzed from different views. First, the GIDAS accident database was used to search for those accidents that would fall within the ODD of the defined exemplary Highway Pilot function. Multiple simulator studies were performed in order to derive an indicator model for human driving performance within selected scenarios of the Highway ODD.

What is the AD (automated driving) capability within the same ODD?

Measuring automated driving capability within PEGASUS was performed through execution and evaluation of scenarios in simulation, on test tracks, or in real traffic. While execution of those tests was performed within subproject 3, this subproject focused on a.) defining a systematic scenario generation process as well as the definition of a scenarios syntax, b.) the calculation of a criticality parameter (KPI) for recorded scenarios and c.) an expert based approach to define automation challenges/risks based on the automated driving system capabilities.

Is the AD (automated driving) capability of the SUT (System under test) socially accepted?

In short, we don't know! We analyzed multiple technology acceptance criteria from different technical systems, such as train traffic, and found a level (or range) of overall performance that is likely to be socially accepted. However, we also found that a real proof of safety can only be given after market introduction and that extrapolating test results or other measures (e.g. extensive test drives) can only serve as an indicator or argument when experience of the final SUT with the ODD is not available.

Which criteria and measures can be deducted from it?

Based on the findings from the previous question, an argumentative structure was created called the "safety argument". Based on this argumentation we are able to argue conclusively for a generally positive risk balance as requested by the German ethics commission. Indicators such as the analysis of accident data, automation challenges or comparisons with the human driver are linked together here and build a combined safety argumentation.

The focus of subproject 2 Implementation Process is the answer to the research question: Which methods, processes, and tools are necessary?

The subproject 2 analyzes existing processes, already established in the automotive industry, regarding the safety argumentation and provides the basis for testing using a modified development process. This leads to a newly extended process methodology. Necessary modifications to the development processes may depend on the level of the automation degree and corresponding ODD. The implementation needs to take into account the step-by-step approach of the automotive development processes and must be sufficiently flexible in order to facilitate the necessary future research and development needs. Nevertheless, it must be sufficiently robust, for the application of functions in the context of a series development, through the inclusion of feedback loops, respectively planes, and with regard to learning effects. The implementation processes analyze the necessity of simulations on various levels. Furthermore, test methods must cover the entire spectrum, beginning with the usage of vehicles on test sites by authorized drivers, through the limited use outside test sites, in development vehicles by test engineers and first tests on some public routes, up to the general tests on public roads.

The subproject 3 Testing evaluates the research questions:

How is the great range of scenarios modeled in scenario-based testing?

At the beginning of the PEGASUS *method* a model for a systematic description of scenarios has been defined with the following six independent layers (figure 1):

1. *road (geometry, ...)*
2. *road furniture and rules (traffic signs, ...)*
3. *temporary modifications and events (road construction, ...)*
4. *moving objects (traffic relevant objects like: vehicles, pedestrian, ...moving relative to vehicle under test)*
5. *environmental conditions (light situation, road weather, ...)*
6. *digital information (V2X, digital data/map, ...)*

Restricting this large parameter space to the operational design domain (ODD) of the test object provides a full test space of the system. This space is not easy for the mental world to grasp. To solve the challenge, in PEGASUS a logical scenario was defined, where some parameters of the scenario model are fixed and some parameters are variable. One example of structuring a logical scenario is to use parameterizable, disjunctive basic constellations of *moving objects* on layer 4, which lead into a collision of the test object with the moving objects when the test object does not intervene. Then, the *logical scenario* describes the complete space of relevant scenario parameters of layer 1 to 3 and layer 5 to 6 and the parameter space of the selected basic constellation of the *moving objects*. So, the entire parameter space of the complete test space is tested by decomposing it with the disjunctive basic constellations of layer 4. Hereby "all" logical scenarios within the space of all logical test cases, which is equivalent to the complete test space, get tested in the simulation.

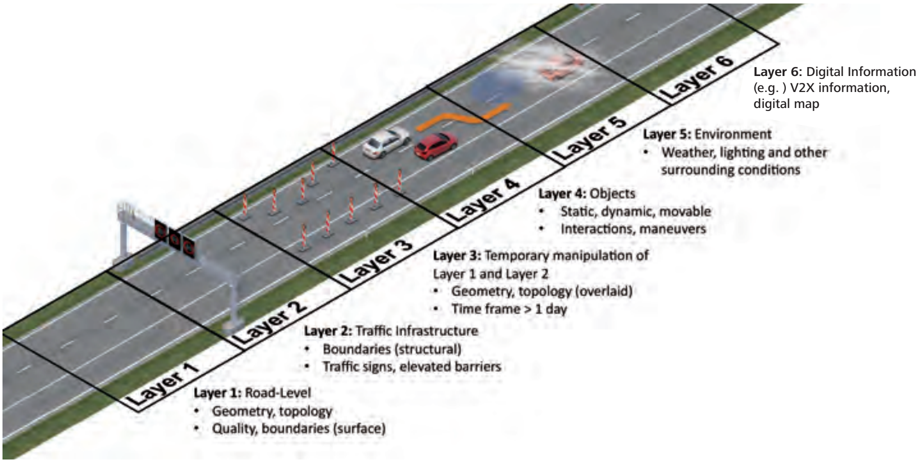


Figure 1: Model for a systematic description of scenarios with six independent layers.

Testing a logical scenario: How does it work?

Within the PEGASUS project, testing the automated driving function within a logical scenario means: safeguarding the test object in the sense of minimizing the risk of collision in the complete parameter space of a logical scenario. The overall goal is to identify collision-relevant, concrete parameter sets or areas in the parameter space to be tested. Due to the large number of tests which needs to be performed, this task is assigned with a sophisticated test concept made up of the test instances simulation, proving gound and field test.

The first task is to formulate the testing as an optimization problem with criticality measure as objective function and pass / fail criteria in its test space. The workflow within the testing of a logical scenario in the framework of the PEGASUS *method* is: sensitivity analysis > optimization > variance/probability based robustness analysis > keeping margin of safety.

With the help of the sensitivity analysis, the variables which contribute most to a possible improvement of the testing goal are identified. Based on this identification, the number of relevant parameters can be reduced dramatically. Empirical studies in the project have shown that in many cases the numbers of most important parameters are between 10 and 20. Due to this, an effective decomposition and effective sample size estimation can be applied. The PEGASUS database provides several evaluations for each logical scenario, which can be used as high quality starting points for the optimization (concrete parameter sets for representative accidents, as well as particularly critical pre-crash constellations observed in reality for the logical scenario) and for the control of the optimization (distributions and correlations of parameters of the logical scenario).

Based on stochastic variation within the parameter space of the logical scenarios, concrete parameter sets are created automatically. Each concrete parameter set corresponds to a concrete test case and vice versa. For each test cases the simulation model is a black box solver and the model responses are evaluated regarding to the pass / fail criteria. The sampling scheme needs to represent the specified parameter distribution and their dependencies with a sufficient accuracy. The research question of the sufficient accuracy is still open. To answer this question, within the PEGASUS project, the approach of creating a transfer function between scenario parameters (input) and test result (output, e.g. collision yes/no, distance between ego-vehicle and relevant target) are evaluated. The approach is also used in other domains.

The robustness analysis is used for investigating the influence of a variation of input variable, e.g. parameter within the concrete parameter set or variation (scatter) of the model response. This could be done variance or probability based.

Following this approach of scanning the parameter space, the test result for simulation is: based on pass / fail criteria evaluated concrete scenarios with probability for collisions. Critical cases, e.g. not fulfilled or close fulfilled pass criteria, get retested in real cars on a proving ground. The target is to execute a detailed search in the parameter space for critical parameter sets.

How can completeness of relevant test runs be ensured? What do the criteria and measures for these test runs look like?

Due to the 6-layer-model of a scenario, the test space of the system and the test space of logical scenarios have been systematically determined and enable a complete description on how to structure scenarios. In this context, complete means that the systematization which describe scenarios is complete due to the possibility to extent the model with additional layers or elements within the layers. The sorting of elements within a scenario to the layers is unique.

But the residual probability for the occurrence of collisions in a logical scenario cannot be given validly, since no parameter set of such a scenario is in the failure range. However, it can be approximated by calculating the probability of parameters of concrete scenarios of increasing criticality that do not represent a collision. During this process a gradient to decreasing probabilities can be seen. Thus, the collisions that occur by crossing the limit to a growing criticality are at most as likely as the previously calculated probability of a previously determined criticality level, e.g. consider the decreasing probabilities of concrete scenarios of a logical scenario for TTC=1.0, TTC=0.5, TTC=0.1, TTC=0.01, TTC=0.001 etc. An upper limit is, for example, the probability for scenarios with TTC=0.01. Since the space of all logical scenarios is equivalent with the complete test space this approach gives an approximation of the residual probability for the occurrence of collisions as well. The scatter of the estimator for the probability of collisions should not fall below a limit (~20%).

What can be tested in labs or in simulation?

In simulation, the test space is scanned for critical cases or areas containing them. The validation is carried out by proving ground and field test.

What must be tested on proving grounds, what must be tested on real road?

Critical cases with not fulfilled or close fulfilled pass-criteria identified by simulation get re-tested in real vehicles on proving grounds. In addition, manually selected concrete test cases can be evaluated on the proving ground (i.e. accident scenarios, rating or certification tests, rare events which can hardly be seen in field tests). The simulation has its limits especially when it comes to areas of extreme driving dynamics and special sensor phenomena that have not yet been implemented in the sensor models. These restrictions are also compensated by testing on proving ground and field tests. Therefore, tests with a high relevance regarding drive dynamics and real sensor performance should be executed on proving grounds. Within field tests it is not possible to test specific test cases. Instead, the behavior of drive features get tested in real traffic. The major target is to find "surprises" (i.e. new scenarios, new parameters). These surprises may be enforced by different guidelines in route (i.e. tunnel) or time (i.e. low sun).

The subproject 4 *Reflection of Results and Embedding* analyses the research questions:

Is the concept sustainable?

The proposal of a Safety Argumentation Structure and the PEGASUS *method* combines several steps to an overall approach which has been defined and refined during the project lifetime. The concept enables further discussions and builds the basis for refinement by successor activities. Internal as well as external feedback shows the need for such a concept proposal, until the final event no other concept has been proposed as an alternative to address the complex question of assessing automated driving vehicles.

How does the process of embedding work?

Successor activities to embed the concept include the work of project partners and upcoming R&D-projects as well as aligning and standardizing activities. The embedding by the project partners has been initiated during the project lifetime but will take some time for each company to complete. A successor project to extend the PEGASUS concept to other use cases as an urban L5 ODD is the V&V-Method project. Further work to use the results for type approval activities has been initiated by the German BASt. Besides these activities a project documentation will be published to enable further exploitation of the results.

2 Overview

The research project PEGASUS (**P**roject for the **E**stablishment of **G**enerally **A**ccepted quality criteria, tools and methods as well as **S**cenarios and **S**ituations on the release of highly-automated driving functions) is government-sponsored by the German Federal Ministry for Economic Affairs and Energy (BMWi). It addresses the research into new methods for the verification and validation of highly automated driving functions (SAE Level 3+ (Society of Automotive Engineers, 2014)). The exemplary test object is a Level 3 highly automated driving function for highways (highway chauffeur).

The test object within PEGASUS the automated driving function is described on a system level. Within the whole project, the test object is handled as a black box. A detailed view of the architecture of the complete vehicle or other single components and their architecture was not in the project focus and was not explicitly handled or tested. This has to be defined through additional system tests. PEGASUS delivers argumentation and corresponding evidence in addition to other items which are needed for automated driving verification and validation. The assessment method provides a concept to improve safety by testing in contrast to safety by design-concepts or similar which are currently not included in the approach. This would be conceivable but needs to be reviewed in further research projects.

In order to define a new state of the art for the verification and validation of automated driving functions, analysis is performed on different test methods, quality criteria, traffic scenarios, tools, and guidelines in four subprojects within the project. Results of the four subprojects contain high interdependencies on each other. Thus, the research results of the various different subprojects were combined in an iterative process to define a common PEGASUS *method for the assessment of Highly Automated Driving Functions*, shown in Figure 2. This overview of the processing chain and interfaces between the different elements of the method provides the starting point for every detailed contemplation.

This chapter describes an overview concerning the architecture of the PEGASUS *method*. The focus is set on the main goals of the 21 different elements of the method and the connections between these elements.

The right side of the PEGASUS *method* describes the process to create evidence for the verification and validation with all steps and interfaces in-between (1) – (20). On the left side, the safety argumentation is located (21). This argumentation (left side) will be compared with the result of the evidence process (right side) at the end of the PEGASUS *method* in order to create a contribution for the safety statement related to the driving function or test object. This can be used for the overall release recommendation.

The process flow of the overall method is read counterclockwise from bottom left to upper left and consists of five basic elements for the verification and validation of the highly automated driving function:

1. Definition of requirements
2. Data processing
3. Information storage and processing in a database
4. Assessment of the highly automated driving function
5. Argumentation

Within these basic elements all relevant methods for the verification and validation of the highly automated driving function are clustered in five sequentially executed process steps. Every usage of the PEGASUS *method* will use these process steps one by one.

The first element is the *data processing*. The input information is comprised, in particular, of the given use case (an item definition of the test object) and existing results of previously executed loops of the PEGASUS *method*. The first goal of this process step is to identify systematically logical scenarios related to the test object based on abstract knowledge. These are transferred directly into the database. The second goal is to convert existing recorded scenarios to a common format. This step is necessary in order to use the different types of information sources within the database. The methods of data processing are described in detail later. The outputs of this processing step are logical scenarios and the information from previously executed scenarios in a common format.

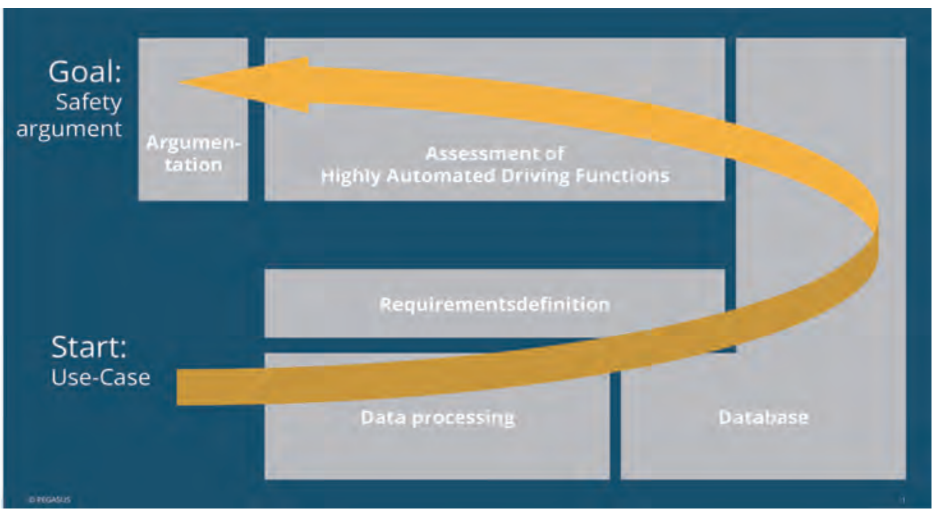


Figure 2: Overview of the PEGASUS *method*

The second element – *definition of requirements* – will be executed parallel to the previously described one. The inputs are, as before, the given use cases, an item definition or results of a previous loop of the PEGASUS *method*. Inside of the second element, abstract knowledge is used to define requirements for the automated driving function or general behavior requirements for the test object. These requirements are used in the database to add evaluation criteria to scenarios to combine them into test cases. Additionally, the identified requirements are used to define process specifications for the fourth element: *assessment of the highly automated driving function.*

The third element is the *database*. Within this element, the prepared datasets in a common format from the first element are used to assign the information to predefined logical scenarios. Furthermore, the prepared datasets are used to define parameter spaces for different scenario parameters. Using this information, the database creates a space of logical test cases with integrated pass and fail criteria from the second element for the different logical scenarios.

In the fourth element, *the assessment of the highly automated driving function,* is executed. The logical scenarios from the database are executed in the simulation and later validated on proving grounds. Systematic field tests will also provide additional findings. The results of the test execution are compared to the pass and fail criteria in order to evaluate them. They are used for a risk assessment to define a safety statement.

Within the last element, the generated evidence is compared with the predefined *safety argumentation*. The comparison is executed in an external procedure model.

2.1 Detailed Description of the Elements of the Method

The previously named basic elements, consist of different kinds of content (data), shown in Figure 3 as data container, and process steps (procedure), shown in Figure 3 as rhombus. The data containers include different kinds of information within the PEGASUS *method,* created by the various process steps. In each case, they include the results from the previous process step and are therefore the main basis for the next step afterwards. These containers include e.g. measurement data, scenarios, test data, etc. Within the process steps, different methods are used to create new output information based on the input, e.g. a reconstruction of accident information.

Every usage of the PEGASUS *method* will use these process steps one by one. In the following, all steps of the method are briefly described.

1. DATA PROCESSING (1,2,4,5)

The PEGASUS *method* starts with the process step of *data processing* with a given use case, as well as all existing logging data and abstract *knowledge,* which are related to the use case or the test object. The process step of data processing is executed in two ways. On one side, abstract *knowledge* (1) is used in a *systematic identification of scenarios* (4) to define *logical scenarios* (9). On the other side, measurement *data* (2) is prepared in the process step of *preprocessing / reconstruction* (5) to create data in a common format (7) for the database.

In most cases, the *knowledge* is represented in an abstract form like text, which cannot be directly used in a technical process. Thus, reviews or technical preparations of the abstract *knowledge* are necessary in order to use this information in technical processes. Within the PEGASUS project, laws, standards, and guidelines are analyzed to define scenarios for the test object in the data processing step. Standards and guidelines are used, e.g. to define minimal and maximal allowed values for different scenario parameters. Furthermore, these information sources are used to create a common understanding of the operational design domain (ODD). This is reasonable, especially if the test object should be released in different countries. The abstract *knowledge* is therefore a valuable source for describing and defining scenarios with a knowledge-based systematic approach. This will be carried out within the following process step: *systematic identification of scenarios* (4).

The process step *systematic identification of scenarios* (4) uses the abstract *knowledge* (1) from the previously mentioned data container as an input source to systematically generate scenarios for the database. Within the project, different methods are applied for these process steps. Examples are the ontology-based scenario generation, which uses the knowledge of guidelines to combinatorically determine all possible scenarios. Another example is the identification of automation risks. Thereby, new scenarios can be found, which are causes of the introduction of the automated driving function. Both methods directly generate *logical scenarios*. Another analyzed way to identify logical scenarios is an expert-based approach, which identifies scenarios around the test object, which causes accidents. With the process step of the *systematic knowledge-based identification of scenarios,* the PEGASUS *method* has the possibility to find a huge set of possible scenarios. In the case of changing *knowledge*, e.g. new or updated laws, it is necessary to generate an updated set of possible scenarios. To achieve this, it is necessary to generate these in an automated way. Therefore, this is a valuable extension of a data-driven method of the PEGASUS, which will be explained next.

Next to the abstract *knowledge*, another source of information is existing measurement *data*, which is used for a data-driven approach within the PEGASUS *method*. In the project, the *data* are based on real test drives, simulations, simulator studies, field operational tests (FOT), naturalistic driving studies (NDS), or accident data. These sources include different kinds of information and are used in different ways in the PEGASUS *method*. For example, accident data is used to generate a value of effectiveness for the automated driving function related to human drivers inside of the ODD. With naturalistic driving studies (NDS), a comparison between the automated driving function and human drivers in different scenarios is possible. Test drives, FOT or random simulations are previously executed scenarios with recorded data. Thus, the use of these information sources describes a data-driven approach to define scenarios for the verification and validation. The challenge with the different kinds of data concerns the different formats and especially the different kind of representations of data. It is therefore necessary to convert the data into a common format within a following process step in order to use the data inside of the scenario database. In the PEGASUS *method*, this will be done in the following process step *preprocessing / reconstruction* (5).

The process step *preprocessing / reconstruction* (5) is used to convert previously recorded data into a common input data for the database. Therefore, it is necessary to define a common format, which includes all required data for reconstruction of the scenario in the database. The format is explained later in the subsection concerning the database. After defining the input format for the database, the data from the different sources, e.g. recorded data from different OEM's, has to be converted to the common input format. This can also include a model-based reconstruction, when a direct conversion is not possible. After this step, the data from every available data source with recorded data can be converted and processed in the database.

In summary, the process step *data processing* has two main goals in the PEGASUS *method*. The first goal is the systematic generation of logical scenarios, which are incorporated directly into the database. The second goal is the preprocessing and reconstruction of recorded data or previously executed scenarios to a common input format for the database.

2. DEFINITION OF REQUIREMENTS (1,3,6)

The step *requirements definition* of the PEGASUS *method* is divided into three elements: a data container, which contains *knowledge* (1), process step with *requirements analysis* (3), and a data container including *process guidelines* + *metrics for HAD assessment* (6).

The data container of the abstract ***knowledge*** (1) is the same container as explained in the previous section. Additional to the already mentioned information sources, laws and regulations of the countries, where the test object should be released, relevant standards (e.g. ISO 26262 (International Organization for Standardization, 2009), SOTIF (International Organization for Standardization, 2019), …) or results of an ethic committee (Ethics Commission Automated and Connected Driving appointed by the German Federal Minister of Transport and Digital Infrastructure, 2019) are used as information source to define requirements for the test object. Similar to the previously explained step of data processing, the abstract *knowledge* cannot be directly used in a technical process. The preparation of the abstract *knowledge* to a technical usable format is carried out in the next presented process step *requirements analysis* (3).

The process step *requirements analysis* (3) uses the abstract knowledge to define technical useable requirements for the automated driving function or general test object. Therefore, in the PEGASUS *method*, two methods are currently applied. On the one side, an approach is analyzed, which is based on social acceptance criteria. For this, the social acceptance from other domains, such as railway or nuclear power plants, is examined to evaluate a possible transfer of these domains. The goal is to find or define similar requirements for the automated driving function from those established technologies. On the other side, an evaluation with a risk-based approach is executed. Therefore, different approaches for the calculation of the risk are applied. One example is an approach which estimates the driving requirements of every scene to handle these scenes. The results of this process step are the definition of different requirements and proving guidelines, which can be used in the following technical process steps.

The evaluated *requirements* of the test object are stored in the following data container (6). This container includes the metrics, based on the analyzed abstract *knowledge*, for the HAD assessment and process guidelines for the following process steps. On the one hand, the results are used in the next step inside of the database to integrate pass and fail criteria into the different scenarios. On the other hand, the results are used as guidelines for following process steps, such as the assessment of the automated driving function.

In summary, the *requirement definition* of the PEGASUS *method* has the overarching goal of defining the *requirements* for the automated driving function based on abstract *knowledge* of different kinds. The results are process guidelines and metrics for the HAD assessment.

3. DATABASE (7,8,9,10,11)

The element *database* consists of three data containers and three connecting process steps between these data containers. Input data for the database are the results of the previously executed elements *requirement definition* and *data processing*. One goal of the database is to handle all collected measurement data from different information sources. Furthermore, another goal is to create a space of logical test cases, based on *logical scenarios*, for the test execution, which is the major output for the following element *assessment of highly automated driving function.*

The database uses as input the preprocessed and reconstructed measurement data, which was converted in the previously executed process step (5) to the PEGASUS data format. The format includes information concerning, amongst others things, the state of the ego vehicle, surrounding objects, and lane information, such as curvature, lane width, etc… To apply metrics and statistics on the measurement data within the database, the PEGASUS data format defines a number of signals and coordinate systems to describe the processed measurement data in a common format. For a further use of the data inside of the database, it is necessary that all signals fulfil a requirement, such as a synchronized sample rate.

Within the database, different metrics are applied to map the measurement data, in the PEGASUS format, to predefined logical scenarios, such as lead vehicle challenger. Hence, the mapping metrics split the measurement data into different time snippets and sort them into the logical scenarios. The time snippets are also used to extract minimal and maximal parameter values to describe parameter ranges within the logical scenarios based on real measurement data. In addition, parameters for the description of stochastic distributions of scenario parameters are extracted. Therefore, it is possible to parameterize the predefined scenario and the scenario from the *systematic identification* with real existing parameter distribution and ranges based on the different sources of information (2) for the following process steps in the PEGASUS *method.* Furthermore, metrics for calculating the criticality for single scenes are applied and stored within the respective logical scenarios. Additionally, the metric mentioned above for calculating the requirements for the scene is applied here to add the information to the *logical scenarios*.

The results of the previously executed process step (8) are stored in the data container for *logical scenarios* (9). Sources for the logical scenarios are, firstly, the results of the *systematic identification* of scenarios and, secondly, predefined scenarios based on expert knowledge. A logical scenario represents a model to describe the environment around the test object with parameter ranges and distribution for the describing parameters. These logical scenarios are described on the basis of a six layer model (road, infrastructure, temporary influences, movable objects, environment conditions, digital information). It also explains the possibilities for describing the behavior of the movable objects, the interaction between on another, and the technical conversion to the formats OpenDRIVE and OpenSCENARIO. The conversion of the scenario to these formats is required to execute the scenarios, e.g. in the simulation in further process steps.

In the next step of the PEGASUS *method* within the database, a space of logical test cases is generated which includes pass/fail criteria and additional preparation to prepare the existing data for a test case derivation. A logical test case therefore includes the logical scenario plus evaluation criteria. The evaluation criteria are added to the *logical scenarios* in the process step *integration of pass criteria* (10). These criteria are represented by metrics, such as TTC, THW, etc., and thresholds for the respective metrics. The information for this process step is the results of the *requirements definition*, where different metrics for the *HAD assessment* (6) were defined. Additionally, the threshold values for the different metrics within the test cases are set which are based on the results of the *requirements definition.*

The results of the process steps *integration of pass criteria* (10) are stored within the data container of *logical test cases* (11). This container includes the test cases, which are relevant for the automated driving function or general test object based on all available information sources (1,2). The test cases are stored in the technical formats OpenDRIVE, OpenSCENARIO, and a format for the parameterization of the logical scenario. The metrics for evaluation of the logical scenarios are stored in external scripts for the application in the following process steps. With this information, it is possible to execute the following basic element *assessment of highly automated driving function.*

In summary, the database is one key element in the PEGASUS *method,* because it connects the preparation part, *requirements definition* and *data processing*, of the PEGASUS *method* with the execution part, *assessment of highly automated driving function.* One main goal of the database is the mapping of measurement data to logical scenarios. As a result, these logical scenarios are managed in the database including filter and sorting functionalities based on characterizing keywords. Furthermore, another task is the generation of logical test cases based on logical scenarios and metrics as input for the following process steps.

4. ASSESSMENT OF THE HIGHLY AUTOMATED DRIVING FUNCTION (12, 13, 14, 15, 16, 17, 18, 19)

The process step of the *assessment of highly automated driving function* consists of four data containers and four connecting process steps between these data containers. Input data for the assessment are the results of the previously executed elements *requirement definition* and the database. The main goals of this element are the test case derivation based on the space of logical test case, the test execution, test evaluation and, finally, the release with generated evidence. The process step is supported by the results of the *requirement definition* using process guidelines. The results are the contribution to the safety statement for the final comparison between the evidence and the *safety argumentation*.

The first process step of the *assessment of the highly automated driving function* is the application of the test concept with various *variation methods*. The inputs for this process step are firstly, *logical test cases* (11) and, secondly, the process instructions from the data container *process guidelines* (6). Within the process step, different *variation methods* are applied to convert the logical test cases to concrete executable test cases. This means that the variation methods are using, for example, stochastic algorithms to select concrete values from the parameter distribution and ranges to create concrete test cases with concrete values for every scenario parameter. This enables the possibility to execute the single test cases on different test tools, such as simulation, proving ground, or real world tests. Additionally, the test concept distributes the single test cases to the different afore just mentioned test environments.

The output of the previous process step are the single concrete test cases. In the PEGASUS *method,* the concrete test cases are represented within the data container (13) by the technical formats OpenDRIVE, OpenSCENARIO and scripts for the metrics to evaluate the *test data*. The parameter ranges and distributions are replaced by concrete values for every single test case. The test cases are executed in the next process step *test execution.*

In the next process step, the concrete test cases are executed in the simulation and on proving grounds (15) supplemented by real-world tests. For this purpose, requirements for the test execution are identified, which are caused by the test of the automated driving function. As a result, different simulation environments are improved based on the identified requirements. Furthermore, different kinds of simulation models, such as sensor or traffic models are developed and integrated within the test execution in order to obtain more realistic test data from the simulation. For a simple exchange of simulation tools and models, new interfaces, such as the Open Simulation Interface (OSI), are used. Similarly, new tools are developed for the test on proving grounds to obtain a more deterministic test execution. Examples are the direct link from simulation to the proving ground infrastructure to compare both results or a

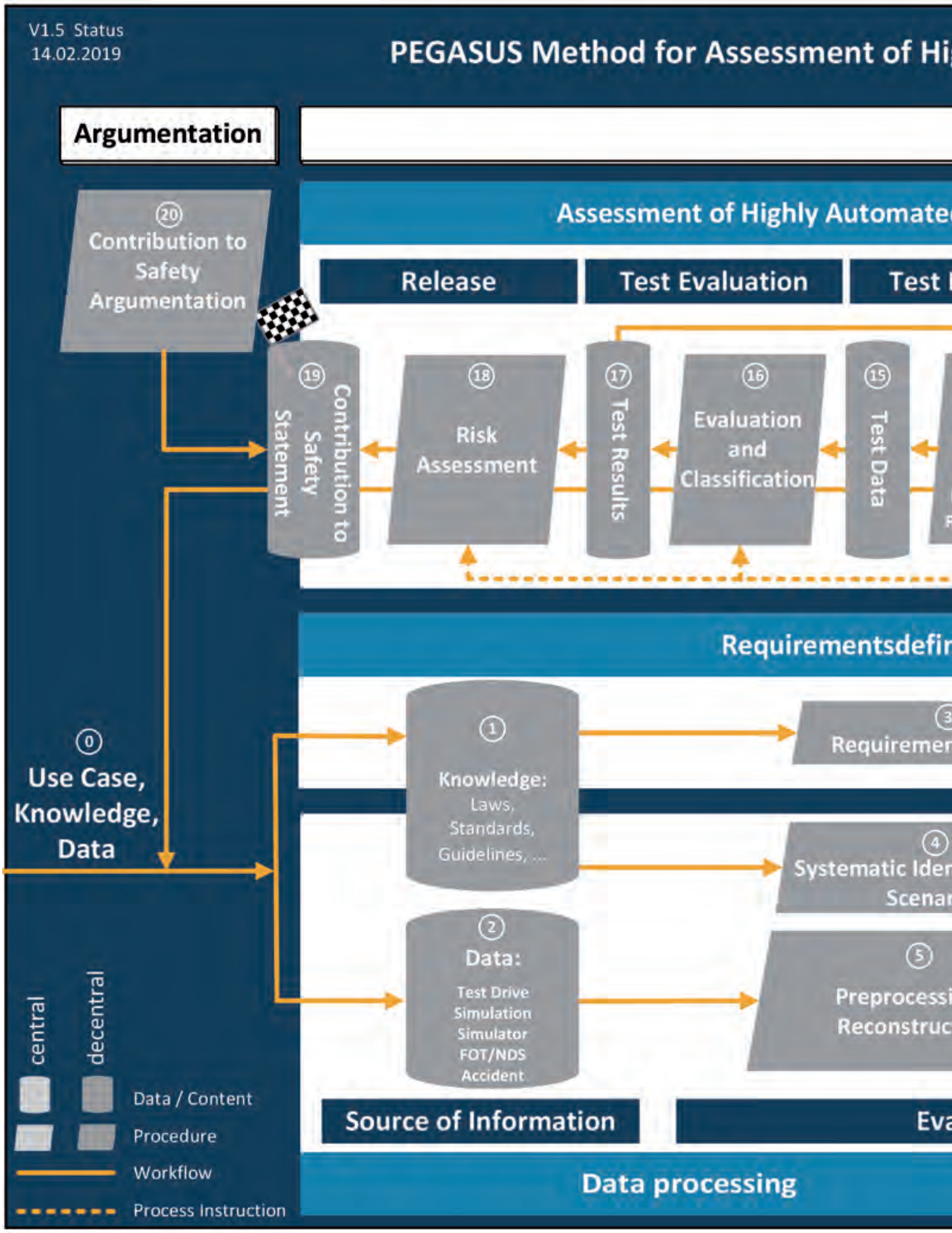
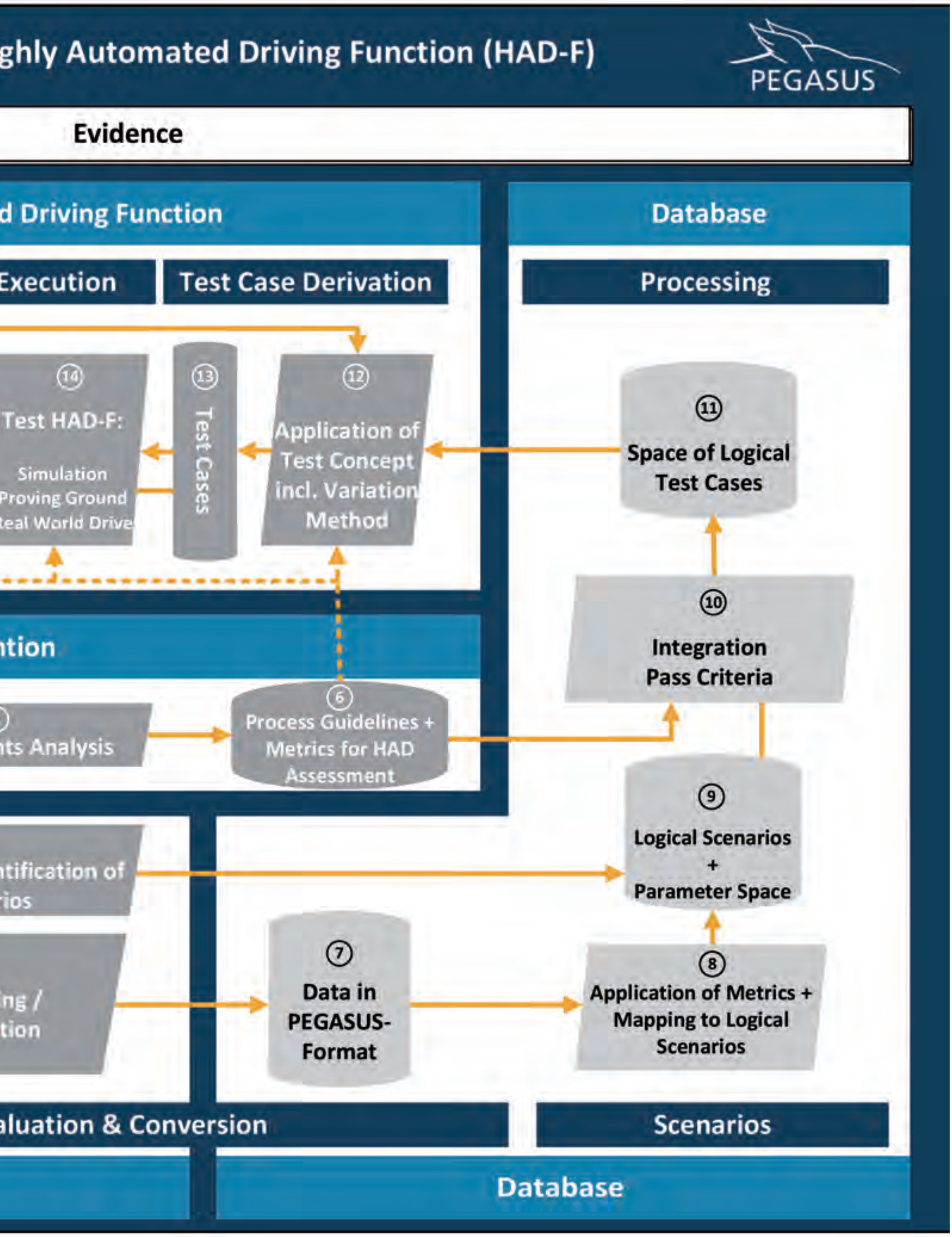


Figure 3: The PEGASUS *method*



ghly Automated Driving Function (HAD-F)
PEGASUS
Evidence
d Driving Function
Database
Execution
Test Case Derivation
Processing
14
Test HAD-F:
Simulation
Proving Ground
eal World Drive
13
Test Cases
12
Application of Test Concept incl. Variation Method
11
Space of Logical Test Cases
10
Integration Pass Criteria
tion
ts Analysis
6
Process Guidelines + Metrics for HAD Assessment
9
Logical Scenarios + Parameter Space
tification of
rios
ing /
tion
7
Data in PEGASUS-Format
8
Application of Metrics + Mapping to Logical Scenarios
aluation & Conversion
Scenarios
Database

new generation of traffic simulation vehicles. These are real vehicles, which drive automatically on proving grounds, based on predefined trajectories. Thus, it is possible to test various scenarios in a deterministic way. Another kind of test execution are real-world tests, where the test object is tested in real traffic. In this case, the execution of the concrete scenario is not directly possible. Hence, the PEGASUS *method* will provide indications for the test conditions to execute the test nearly on the concrete scenario.

The results of the test case execution are stored in the data container *test data* (15). The *test data* describe the information concerning the test execution in the form of traces of signals. These signals have the same format as the data in the PEGASUS format (7). Therefore, it is possible to use the *test data* as additional input for the PEGASUS *method.*

In the next process step (16) of the method, the *test data* are evaluated with different metrics, such as TTC or distance check, and categorized in groups, such as collision. Based on the test results, the process step starts an iterative assessment with two purposes. On the one side, the test results are passed back to the stochastic *variation* (13) in order to find more critical scenarios in the parameter space of the logical scenario based on the last test results. On the other side, the test results are used to identify concrete scenario, which are used for a cross-verification on other test instances, such as proving ground tests. If the results fulfill certain criteria, the iterative assessment process breaks up and the evaluated test results are stored in the following data container for the test results.

The data container *test results* (18) contains the evaluated results of the different test runs. The test results are described as traces of recorded signals of the process of the test execution. Additionally, the results of the metric evaluation are added to the traces. Furthermore, it is possible to tag timestamps with maximal and/or minimal vales of the different metrics. The *test results* are used in the following process step for a *risk assessment.*

The process step of the *risk assessment* (18) applied an additional evaluation to the individual executed test cases. The difference between the previously executed test *evaluation* in process step (16) and this process step is the focus of the evaluation. In contrast to the last *evaluation*, the purpose of the evaluation is to confirm the compliances of the test object with pre-defined behavioral criteria. Therefore, the individual test cases are evaluated with other metrics as in process step (16). In the specific PEGASUS context the pre-defined criteria are, e.g., keeping appropriate safety distances, not causing collisions and, if possible, mitigating collisions, as they comply with the test concept. During the risk assessment, for each of these criteria, an evaluation we made as to whether the HAD-F had complied with it or not. Based on the result for each criterion, a method is proposed to decide if a single test-case has passed or failed in relation to the pre-defined behavior criteria. In the PEGASUS *method*, the results of the risk assessment are stored in the data container with the *safety statement*.

5. ARGUMENTATION (20)
The last step in the PEGASUS *method* is the *application of the safety argumentation together with the generated safety statement.* The PEGASUS *safety argumentation* is to be understood as a conceptual framework to support securing and approval of higher levels of automation through structure, formalization, coherence, integrity and relevance. It is structured by introducing five layers. Established formalizations, such as goal structure notation, are used wherever possible in order to describe each layer's elements. Those elements are linked across the layers in order to form a coherent argumentation. The evaluation of each elements' integrity in order to establish a reliable safety argumentation is also suggested. The central assumption of the PEGASUS *safety argumentation* is: If a chain of arguments, which was created taking into account the proposed framework of the PEGASUS *safety argumentation*, stands up to a critical examination, this will support securing and approval of higher levels of automation.

2.2 Summary

In summary, the PEGASUS *method* describes a systematic concept for a scenario-based verification and validation approach for automated driving functions. With the central elements of the method of *definition of requirements, preparatory data processing, information storage and processing in a database, assessment of the automated driving function and finally the safety argumentation,* the PEGASUS project takes a first step from a distance-based approach with random testing character to a systematic scenario-based v&v approach. The project, however, also exhibits two essential findings: many questions are still open and new research questions still arise. However, the PEGASUS *method* has the possibility to sort the open and new arising research questions within a big picture or an overall architecture for the next verification and validation activities.

3 Reflection

The following chapter describes reflection of the PEGASUS project. The setup of the consortia, the work model within the project, and the project results are reflected. In addition, the dissemination and exploitation is discussed. Afterwards, an outlook and recommendation for further activities will be given.

3.1 Reflection of Consortia Setup and Workmodel

Across the project runtime it became evident that the verification and validation on the level of safety goals is not only a question of respective OEMs but rather needs a general acceptance and coordination. To ensure this the PEGASUS project supported and promoted a precommercial exchange between OEMs and TIER1 to initiate discussions on topics which were never in the focus of such projects before. Due to the setup of the consortia with not only OEMs and TIER1s, but with small and mediumsized businesses, research institutions, federal authorities and NGOs as well helped to start the interdisciplinary discussion between the project partners. The national discourse has simplified the complex discussions. The focus of all partners was in the direction of a realizable and implementable technical solution for a joint approach for the verification and validation of automated driving functions. In the case of irreconcilable differences the steering board provided a mean for escalation to reach a decision and invited experts for detailed discussions.

But the discussions within the project were primarily of technical nature with the goal for an objective safety validation by testing. A discipline wide exchange with safety, security or legal experts was not planned in the project. In an afterthought it is recommended that for further research projects on the topic of verification and validation, an integration of such experts will therefore help to provide a holistic discipline-wide discussion. One solution could be an extension of the PEGASUS *method* to an overall v&v architecture to combine the aforementioned disciplines with the existing PEGASUS v&v activities.

Another point, which arose during the project was the remarkable national and, in particular, international interest in the project. The project was transparently for national/international interests within the corset of a national funded project, but the wish for transparency bound unexpected resources from the PEGASUS partners. One example for this was the unplanned overhead due to duplicate documentation in English and German. It is therefore advisable to integrate a strong focus on the international dissemination in the application phase of further projects.

The work model within the project was structured into four subprojects. This enabled the work on relevant topics independently from each other: processes, system definition and analysis, test execution and result reflection. Additionally, the structure facilitates the generation of expert groups, such as stochastic variation, sensor models or safety argumentation, meetings and events inside of the subprojects. The assignment of different companies and disciplines to subprojects encouraged collaboration and provided also a very high interaction between the partners over company borders. Another positive aspect of the work model became apparent at the half time event. Here the current existing results were presented in booths organized and structured in PEGASUS topics and not company booths. Thus, the expert groups had the possibility to present their research results in the form of a common project approach without company branding.

On the other hand, the predefined subproject structure did not support a consistent workflow between project parts over the whole project lifetime. In a public funded project work packages of all partners are predefined with the necessary manpower and expected results during the project starting phase and fixed in a project description (VHB). During the evolving project it became necessary to readjust the original strategy in several cases. Some were necessary because of different interpretation in the working packages by partners but most of them were needed because of new findings. Due to the highly interconnected working packages these changes lead to noteworthy shifts in corresponding working packages and budgets of some partners. This provided a certain challenge not only for the project management but especially for smaller companies and universities.

In some cases, results or status of expert groups within the subprojects were not as transparent as expected for other subprojects. For example two main interfaces between subproject 1 and subproject 3 lead to challenging discussions. Additionally, subproject 2, analysis of processes, generated a lot of results, which are only partially integrated into the PEGASUS *method*. Finally, the separation of subproject 4 from subprojects 1-3 lead to challenges for the execution of the planned result reflection.

From a pure technical point of view a solution for the outlined challenges could be a more agile approach for the project organization. Other domains demonstrate that projects with high uncertainties in the questions "What to reach" and "How to reach it" do use an agile approach (cf. Stacey Matrix) with supporting tools, such as ticket-, wiki-system, and revision control as organizational form. This could help to react to changes in the project caused by research results or general changes. For a realization of an agile approach within the project, an overall architecture, like the PEGASUS *method*, is necessary at the start of the project in order to sort the work packages into a big picture. In addition to this architecture, a "technical chief product owner" for the architecture is necessary, who would have the task of adapting the architecture in an iterative process when it is necessary. Another important task of such a chief product owner would help the subproject teams with the technical coordination within the project. Unsolved by such an approach would be the legal aspects of a public funded project, where the expected contributions of each partner have to be predefined in a project description at the project start.

3.2 Reflecting the Results of the Project

3.2.1 Overall PEGASUS Method

The PEGASUS *method* shows partial results of the project in an overall architecture. By this method, the complexity of the assessment task of an automated driving function can be indicated and explained. Furthermore, the method depict and express how the necessary elements for the assessment depend on each other and act interconnected in a process chain. In addition this top down approach provides the possibility of stressing the interactions and methods of different elements of the method, whereby missing interfaces or different understanding of elements can be directly detected. On the other hand there remains a certain risk that the assessment task would be oversimplified by not transparent parts such as the enabling and validation activities for tools in this proposed process chain.

The picture of the PEGASUS *method* suggests a linear step-by-step approach. The different blocks, however, have different life cycles and not only adjoining blocks might have dependencies. One example is the safety argumentation and the contribution of the single steps to these safety statement. The database shall be always on and continuously filled with up-to-date information from changing requirements as well as scenario extensions. Whereas the assessment part is timely connected to the assessment during a vehicle development project and will have several instances, the database shall have a common core to reach its targeted determination.

In conclusion, on the highest abstraction level, the figure indicates the important findings of the PEGASUS *approach:*

1. Evidence requires an argumentation.
2. The database is not a static construct but evolving circularly.
3. The output is a space of logical test cases and NOT a list of concrete test cases.
4. The test concept has an iterative/searching character.
5. Some elements of the overall process shall be centralized, others should not.

Currently, the method is validated by an example for a first proof of concept. Additionally, the conformance of the method to existing company processes has been analyzed and the check for the integration is work in progress. The proof of concept for a series application is pending and currently not implemented. Furthermore, the transferability to other use cases (domain, automation level) has not been demonstrated completely. Further projects shall therefore evaluate the PEGASUS *method* for additional use cases, such as other operational design domains or other automation levels.

3.2.2 Requirements

Within the project, different information sources (e.g. guidelines, standards, rules, ethics and social requirements) have been analyzed to identify requirements for the test object (the highway chauffeur). By defining the test object as a black box without detailed functional architecture, such as sense - plan - act, safety requirements had to be defined on a high level. On one hand, this has the advantage that the PEGASUS *method* approach for defining requirements should be applicable to other automated driving black boxes within the defined use cases. On the other hand, this black box approach does not provide component requirements based on a functional system architecture. The *plan* part as a whole was within the scope of the project. Due to this the PEGASUS *method* does not cover lower functional levels or the parts *sense* or *act.* For further projects it could be necessary to use a more detailed function architecture, such as *sense* - *plan* - *act,* as test objects in order to define requirements on a more detailed level inside of the three parts.

A major requirement for an automated driving function would the public accepted error rate, since perfect technical systems cannot be realized. To make a first proposal for acceptable macroscopic numbers of e.g. accidents/fatalities, different information sources in parallel domains, like railway or airplanes, have been analyzed. Currently, a still open challenge is to identify a possibility or approach to connect these macroscopic numbers with singular test case results. To make a step towards a first approach, microscopic requirements have been defined to evaluate test cases (passed/not passed) for a safety statement. To define safety goals, such as no collision, an expert-based approach is applied within the project. An exhaustive validation of safety goals must be evaluated in further projects. The started discussions on safety goals should be continued and extended to a review by national and international community groups.

3.2.3 Data Processing

Within data processing, different systematics, such as ontology and automation risks, and data-driven approaches have been applied to identify relevant scenarios based on different information sources. The state of the art for scenario modeling has been extended to six layers of scenario description. Currently, mainly the 1st layer on road networks and the 4th layer on dynamic objects have been modeled and explored within the PEGASUS *method* with a scenario-based and data-driven approach. To provide a data-driven approach, different information sources were used to convert available data into a common database format. The conversion process generated for all data-providing partners a certain effort due to the application of writing converter, video generation, handling privacy (e.g. GDPR) etc. For further projects one solution to overcome this challenge could therefore be to define data suppliers during the project setup. By this the effort and resources for generating and converting data for a common scenario database should be planned.

Next to the data-driven approach, two additional systematic approaches were developed:

1. Identification of automation risks, these automation risks generate logical scenarios which can also be integrated into the database.

2. An ontology has been modeled and applied to have an alternative approach for defining the content of the scenario layers 1-5. The ontology-based approach generates directly logical scenarios.

The integration of automation risks was implemented. The results seem to be promising, but complete integration into the PEGASUS *method* had not been executed by the end of the project. However, the influence of these scenarios within the tests and the usage for the safety assessment needs to be finally discussed yet.

3.2.4 Database

The database is one central element in the PEGASUS *method* because the database connects the preparatory part with the execution part of the method. To achieve this the database has to manage and handle the exchange of data from different sources provided by all partners. A challenge was that every project partner wanted to work directly with the database input and output at the start of the project, while the database algorithms were still under research and development. For further projects with a central database, it could therefore be important to use these information to define interfaces and example data in an early stage of the project. Work packages based on the data base might start early with focused discussion on those examples.

Next to the data handling, a process was implemented to identify scenarios and their parameter distribution based on the provided measurement data. These data are used for the assessment of the test cases variations. In the database there is also the possibility to tag those scenarios, which are based on automation risks and accidents. An open research question is the identification of unknown scenarios for relevant corner cases, which do not fit into the predefined scenario template yet.

3.2.5 Assessment

The assessment is the main test execution and evaluation part in the PEGASUS *method* with simulation, proving ground tests, and field tests. For the simulation, different approaches of parameter variations in the space of logical test cases have been implemented. First evaluations of the parameter space were applied on selected special test cases. An exhaustive parameter variation with extended parameter sets, such as parameters for multiple sensor models, or with a number of multiple scenarios has, however, not been realized. Additionally, the question as to how the defined test cases generate the required evidence for the defined safety goals has not been explicitly ascertained.

To create more realistic results from the simulation, different sensor models have been implemented. Therefore, formats for describing the scenarios, such as OpenDRIVE and Open-SCENARIO, were developed up to a common standard and transferred to the standard organization ASAM. First approaches for validation of the simulations and sensor models have also been developed, but an extensively validation of the models has not been executed. The proving ground tests have been improved by an implementation of coordinated testing with self-driving traffic simulation vehicles. Here, first steps for an automated adaptation from the simulation formats, such as OpenDRIVE and OpenSCENARIO, to the proving ground are performed. An implementation of a full support of the formats is in progress.

The assessment has shown that many promising approaches have been developed, but the research questions concerning the execution and evaluation of test cases are still open or arising. To handle these challenges of open research questions, further projects should define tandems of project partners for parts of the PEGASUS *method* in order to clearly focus on single research questions.

3.2.6 Safety Argumentation

The last step of the PEGASUS *method* describes the safety argumentation. The known goal structuring notation that formalizes the safety argumentation has been extended to model the individual and societal perception of automated driving safety. The safety argumentation also defines the PEGASUS *method* as one element in an overall argumentation and sorts the PEGASUS activities into the overall picture of a common v&v approach including testing, safety, legal, etc.. For a first application of the safety argumentation, an exemplary argumentation chain was defined. Therefore an international discussion about the structure was started to reach a common approach. The final structure of the safety argumentation is still ongoing due to the fact that the safety argumentation is still a living document. Thus, the document should be improved in an agile process in further national or international projects to apply or develop a common safety argumentation to demonstrate how a predefined safety level (see 3.2.2) can be achieved. The work on the safety argumentation has shown two important findings:

1. the identification and development of the safety argumentation is an additional research topic next to the other research question within the PEGASUS *method*.

2. the safety argumentation can support the project to define a structure for the work package due to the argumentation chain.

It is recommended for further projects to start with the safety argumentation very early in the project to define the contribution of each single work packages to the project architecture and to the argumentation chain of safety.

3.3 Reflecting the Dissemination

Due to the global nature of individual mobility and public traffic, the question of automated driving is not a question of an individual country. Different governments, companies and academic institutes are willing to address existing challenges of today's traffic by automated driving. Nevertheless PEGASUS started the discussion on how to implement automated driving safely on a national level. This happened on purpose, since the partners agreed in the beginning of the project to work for a certain period of time on a national level to achieve very fast first results, which could be used to initiate a more thorough international discussion afterwards in a second step.

Figure 4: Overview of the international project activities

The PEGASUS partners do not see the basic discussion on safety and how to generate the necessary evidence for a safe deployment as a competitive topic. For this reason the PEGASUS partners actively looked for an exchange on selected topics. Beginning with the halftime event in Aachen, which was open to a public audience, four consecutive international workshops on selected topics were held in Germany, Austria, USA and Japan. Besides these symposia, PEGASUS partners presented the status of the project at different conferences and meetings, and to, commissions and interested parties upon request. Many activities as depicted in the figure 4 took place in the US, Europe and Asia.

The feedback PEGASUS received from these activities, was mainly positive as the transparency on a running national project was increased. Challenges of this complex topic as well as solution proposals were highlighted and brought to the international AD-Community. PEGASUS offered concrete input and lessons learned to e.g. starting standardization activities. And vice versa PEGASUS was stimu-

lated by the exchange with international partners. Thoughts on e.g. scenario definitions, safety argumentations, and metrics found on international exchanges where reflected and partially integrated.

Besides these positive effects, the dissemination activities during the project's runtime generated unplanned and not estimated efforts for the project partners. In particular the complexity was increased due to the additional input from outside the project's boarders.

The lesson learned from this activities for future projects is to calculate and expend more effort for the dissemination as a concrete work package within upcoming projects. The PEGASUS project can be used as a good starting point to continue and extend the exchange of an international community for v&v activities regarding automated driving, because this community is not as established as, for example, safety communities. This community should be the basis for international research and standardization groups at the beginning of next projects.

3.4 Reflecting the Exploitation

Usually, the exploitation of research and development projects starts with the results, i.e. at the end of a project. This also applies for PEGASUS as well, however there was and continuously is a huge interest in and need for solutions to the research questions PEGASUS stated and focused on.

For this reason, the project was designed in such a way that from the beginning, results could be embedded into the partner companies / institutions as early as possible. This had positive effects as, for example, the PEGASUS *method* was created to be able to communicate a certain baseline of the activities. Company internal stakeholders had to be identified and were approached actively during the project's runtime. Thereby, an early feedback at the company's level as well as in the PEGASUS-project was generated. Non-PEGASUS perspectives had been requested and collected. The half time presentation as well as the coming final presentation could and will be directly used by addressing the internal and external stakeholders.

For the early exploitation of results, the same applies as for the early dissemination. It generates an overhead that is not directly spent on project work. For this reason, the matter of when and how to perform this exploitation is a balancing act as, on the one hand, the results should be used as early as possible, and on the other hand, the reason for the project was that time and special support through governmental funding are necessary in order to have the time and circumstances for answering open questions.

One lessons learned for future projects is to have exploitation activities that are described generically in the project proposal at the end of a project starting at the final event. This exploitation should be planned with sufficient effort after the project's elaboration within every part project and not as a separate one.

It is for the project partners to decide how to exploit the results and how this finally can be reflected in the future. The outlook we can give (see next chapter) is limited to the national initiatives, being well aware, that other nations, international projects and international companies are driving this topic further and can hopefully can benefit from lessons learned through PEGASUS.

4 Outlook

4.1 An Outlook with the PEGASUS Perspective

PEGASUS has a pioneer role in the topic of V&V research for SAE L3 and higher levels, as it was the first project solely focused on V&V topics. The funded projects itself will terminate at the end of June 2019. Although the research questions have been answered initially further activities that are related to and succeed PEGASUS are on their way.

Originating from the VDA lead initiative on automated driving are several proposals addressing to the respective ministries for funding and enabling national research and development projects. The main motivation of all these projects is to enable a safe and higher automated mobility to address the challenges of future transport. And again, these projects are in the pipeline either already granted or in proposal state and aim to work on the enabling methodologies, tools and frameworks for these upcoming technologies.

SETLevel4to5 began in March 2019 and focuses on a simulation platform to increase the efficiency of testing automated driving systems. It is based on the simulation environment of PEGASUS and is extending the use case from highways (PEGASUS) to other areas (e.g. city intersections) as well as the SAE level of automation from Level 3 (PEGASUS) to higher levels.

The project *Verification & Validation (V&V) Methods* is in the proposal phase but aims to perform the analogous extension of the PEGASUS *methods* like in the way that *SETLevel4to5* is doing for simulation. A main difference between *V&V Methods* and PEGASUS is the additional focus on vehicle system components such as perception and planning components. The promise of also evaluating not only the overall system but a decomposition is to increase efficiency by these modular v&v activities.

Another important topic to enable a safe and state-of-the-art automated driving system is to handle advanced AI within an automated vehicle. The project "KI-Absicherung" will strongly focus on this research question.

All these activities have in common the fact that mutual databases would be of benefit or even necessary. A database that addresses the scenario modeling and collecting features has been initially developed in PEGASUS. However, if this database should also be reliable in the context of a type approval, it seems reasonable to establish a sovereign structure to operate, maintain and extend such a database.

4.2 The PEGASUS recommendation

PEGASUS has aimed for an exceptional goal. As the first research and development project in the automotive industry (known to us), PEGASUS aimed for a common and acceptable approach to verify and validate an automated driving functionality. The validity of such an approach cannot be proven before the first vehicles have been brought their intended use. However, PEGASUS has proposed answers to essential questions. It is thus the task of the community to analyze these answers, to apply them or to falsify and improve the proposed approach.

Our recommendation is to establish a community who set this as their No.1 goal: Try to falsify existing and promote new proposals and thereby continuously improve the AD V&V approaches. Therefore, funded projects are good starting points. Additionally, the involved companies must not see safety as a competitive area and correspondingly, shall enable their employees to exchange and to improve an international approach.

An accepted safe and efficient mobility with the suitable level of automation will be the result.

5 List of References

Ethics Commission Automated and Connected Driving appointed by the German Federal Minister of Transport and Digital Infrastructure, 2019. BMVI. [Online]
Available at:
www.bmvi.de/SharedDocs/EN/publications/report-ethics-commission.pdf?_blob=publicationFile

International Organization for Standardization, 2009. ISO/DIS 26262-1 - Road vehicles - Functional safety, s.l.: International Organization for Standardization / Technical Committee 22 (ISO/TC 22).

International Organization for Standardization, 2019. ISO/PAS 21448:2019 - Road vehicles - Safety of the intended functionality, s.l.: s.n.

Society of Automotive Engineers, 2014. Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems. Surface Vehicle Information Report, Issue J3016.

Wachenfeld, W. & Winner, H., 2015. Die Freigabe des automonen Fahrens. In: M. Maurer, J. C. Gerdes, B. Lenz & H. Winner, Hrsg. Autonomes Fahren. Heidelberg: Springer Verlag, pp. 439-464.

6 PEGASUS Project Partner

- Audi AG
- ADC Automotive Distance Control Systems GmbH
- BMW Group
- Continental Teves AG & Co. oHG
- Daimler AG
- Forschungsgesellschaft Kraftfahrwesen mbH, Aachen (fka)
- German Aerospace Center (DLR)
- iMAR Navigation GmbH
- IPG Automotive GmbH
- Opel Automobile GmbH
- QTronic GmbH
- Robert Bosch GmbH
- Technische Universität Darmstadt – Automotive Engineering (FZD)
- TraceTronic GmbH
- TÜV SÜD Auto Service GmbH
- VIRES Simulationstechnologie GmbH
- Volkswagen AG

Contact:
PEGASUS Project Office:
Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR)|
German Aerospace Center
Institute of Transportation Sytems | Dipl. Ing. Jens Mazzega
Lilienthalplatz 7 | 38108 Braunschweig | Germany
Phone: +49 531 295-3456 / -3401
Mail: info@pegasusprojekt.de | www.pegasusprojekt.de