Efficient test methods for the system test of highly networked safety systems

Kathrin Sattler, Andreas Raith, Thomas Brandmeier

Field of Expertise: Automotive Mechatronics
Institute for Applied Research (IAF)
Ingolstadt University of Applied Sciences

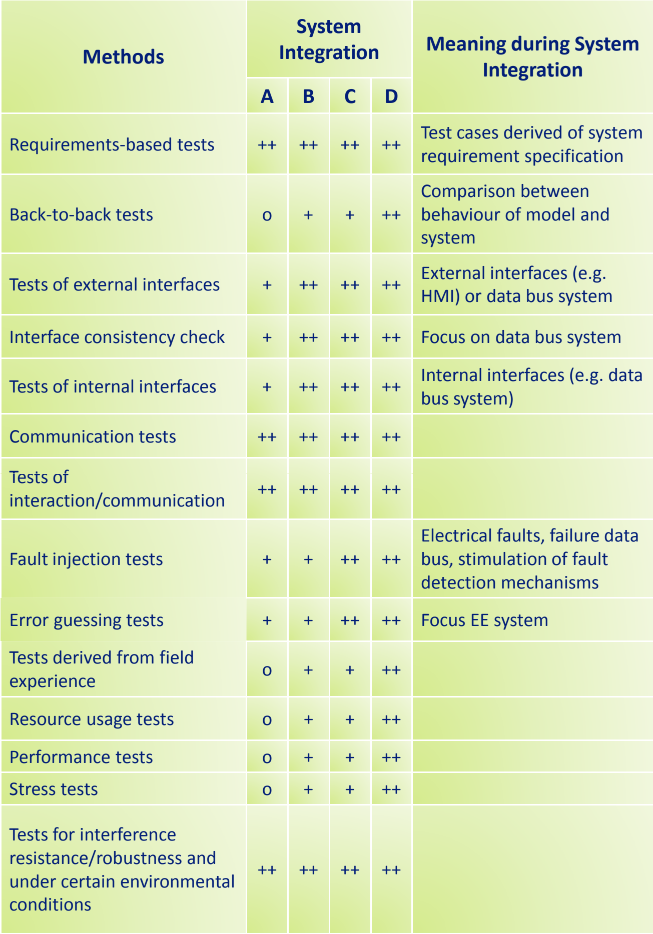
Email: {kathrin.sattler, andreas.raith, thomas.brandmeier}@haw-ingolstadt.de

***Abstract*—Safety systems in the automotive field were developed separately for a long time. Nowadays active and passive safety systems are networked, exchange information and rely on each other. This leads to an increasing complexity of the communication channels and numerous control unit functions and variants. Therefore the development and test process is getting more and more complex and unmanageable. Also testing always means a compromise of duration, quality and costs. Despite those challenges a high error detection and therefore reliability shall be reached. Due to those changes especially the system test where different components interact for the first time and many functions that rely on collaboration of different elements can be tested for the first time in the development cycle is a big challenge. Here arises the potential to improve contemporary test methods and strategies and to examine new ones and to involve them in existing processes. This paper describes one possible approach for a combination of test methods leading to an efficient test strategy for the system test of the airbag control unit.**

I. Introduction

In the last years the complexity of the networking of different control units in vehicles increased rapidly. Different bus systems like FlexRay [5], CAN [10] or LIN [11] do not only communicate with each other directly, but also via a gateway. Therefore the whole vehicle is interconnected and a lot of information is exchanged. Data consistency is crucial since the control units rely on information of other control units, e.g. the airbag control unit (ACU) uses the four wheel speeds of the electronic stability control. Here arises the necessity for complex tests for ensuring the functionality and safety of the functions throughout the whole development cycle of a control unit. The problem here is that especially during the system test where a control unit is embedded in the whole mostly simulated peripheral control unit composition and is tested for initial system requirements it is not possible to cover all possible error combinations due to the enormous number of combination variants. Therefore the absence of errors can never be proved entirely. However in safety relevant fields such as the development of airbag control units errors must not be present. For this reason efficient test methods and strategies are crucial.

The standard ISO 26262 Road vehicles - Functional safety [7] which was published in 2011 provides measures, activities and methods for development and test of functional safety relevant electrical and electronic solutions for control units in vehicles up to 3.5 tons. Figure 1 shows the suggested test



Methods	System Integration				Meaning during System Integration
	A	B	C	D	
Requirements-based tests	++	++	++	++	Test cases derived of system requirement specification
Back-to-back tests	o	+	+	++	Comparison between behaviour of model and system
Tests of external interfaces	+	++	++	++	External interfaces (e.g. HMI) or data bus system
Interface consistency check	+	++	++	++	Focus on data bus system
Tests of internal interfaces	+	++	++	++	Internal interfaces (e.g. data bus system)
Communication tests	++	++	++	++	
Tests of interaction/communication	++	++	++	++	
Fault injection tests	+	+	++	++	Electrical faults, failure data bus, stimulation of fault detection mechanisms
Error guessing tests	+	+	++	++	Focus EE system
Tests derived from field experience	o	+	+	++	
Resource usage tests	o	+	+	++	
Performance tests	o	+	+	++	
Stress tests	o	+	+	++	
Tests for interference resistance/robustness and under certain environmental conditions	++	++	++	++	

Fig. 1. This figure shows the test methods for the system test of vehicle electrical and electronic systems suggested by the standard ISO 26262 distinct by the different Automotive Safety Levels (ASIL) [7]. While a number of possible test methods are recommended it is not provided how those methods shall be combined and to what extent they should be used.

methods for the system test in ISO 26262. Here most of the already used test methods are collected and given a priority for different ASIL levels. This is only a recommendation, but in case of an error in the vehicle that leads to injuries or even death legal questions for the manufacturer arise. Therefore to be sure that such errors are expunged a verification for sufficient testing has to be given. While ISO 26262 shows a number of possible test methods it does not provide a recommendation how those methods shall be combined and to what extent they should be used.

Figure 2 shows the test activities in general. Especially the identification of test cases is an important aspect which requires much knowledge. Most of the time testing of single errors does not uncover errors, but the combination of errors. Testing of the complete combination of all test cases is not possible due to the enormous number. Often errors are only detected in the vehicle driving tests after the in-office-test-phase when all components work together in the real vehicle. The detection of errors in this state that result of problems in the specifications and implementation is of course too late in the development cycle due to the high costs and high effort for error removal. Therefore efficient test methods and an elaborated test strategy is needed to find errors early in the development process. This paper describes one possible approach of combining different test, execution and evaluation methods in order to contribute an efficient test strategy to solve challenges for the system test of highly connected safety systems. The system test is chosen as point of action since here for the first time the control unit comes into its later environment and many problems occur for the first time. But since the system test is mostly done in-office, it is not too late in the development cycle to eliminate errors. The system test covers the testing of requirements therefore we are talking about black-box-tests here.

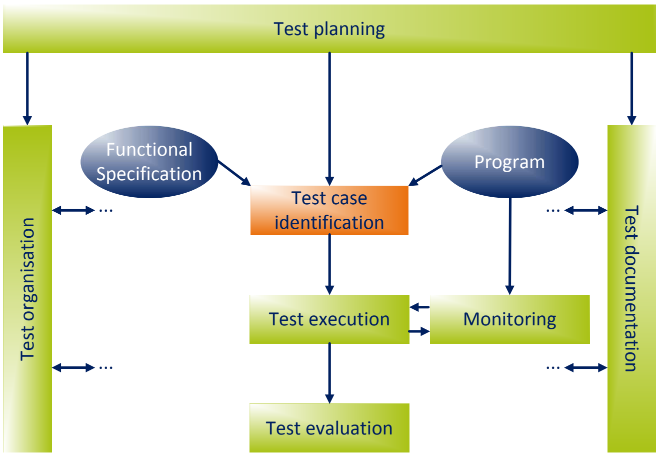


Fig. 2. The general test activities consist of overall test planning, the columns test organisation and documentation and the main part test case identification which depends on the Functional Specification and the Program. Resulting out of the test cases are the test execution with monitoring and the evaluation. Critical, essential and most difficult is here the test case identification [21].

II. State of the Art

Testing in the lifecycle of a control unit is done in various stages and in different ways. Figure 3 shows the well-known V-cycle which represents the development lifecycle throughout the whole process. It includes some test methods that are used in the different stages of development. Early in the project models are used to create the algorithms and structure of the system. Here Model-in-the-Loop (MiL) is used for testing in a simulation environment in order to find mostly algorithm errors in an early stage. After implementation Software-in-the-Loop (SiL) is used as a measure to execute the code even before hardware is available. Therefore many coding errors can be found early in the development cycle. As soon as hardware prototypes are available Hardware-in-the-Loop (HiL) is used to different extents. HiL tests can range from just having the hardware of the control unit itself while the rest of the periphery is simulated to having a composition of several control units, bus systems, sensors and other peripherals. Often faults are injected trace-based which means that recorded or artificially created signals for busses or sensors are inserted into the system over the real interfaces of the control units. Deriving test cases can also be done by model-based testing. But since in our case no model is existing, this shall not be considered here. Later in the development cycle real vehicles are used for test drives and long term tests.

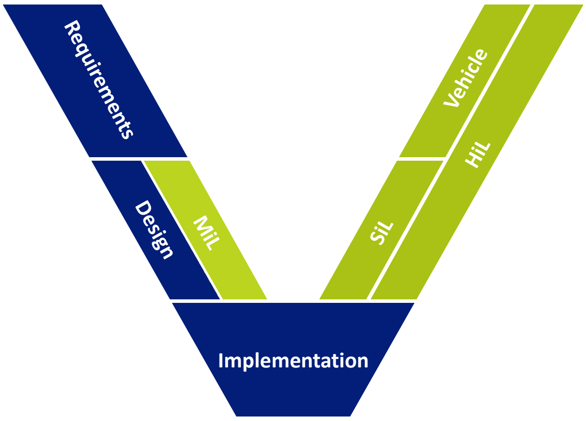


Fig. 3. The well-known V-cycle represents the development lifecycle of a control unit throughout the whole process. Also shown are different test methods that are used in different stages of the development. After the requirements are defined the design is verified by MiL tests. After implementation SiL and if hardware prototypes are available HiL tests are executed. In the end of the development cycle vehicle test drives are conducted.

As Figure 4 shows the network of control units in a contemporary vehicle is already very complex and consists of many individual components. Different bus systems connect many different control units and share information via a gateway. A diagnose interface can provide information about errors for the garage.

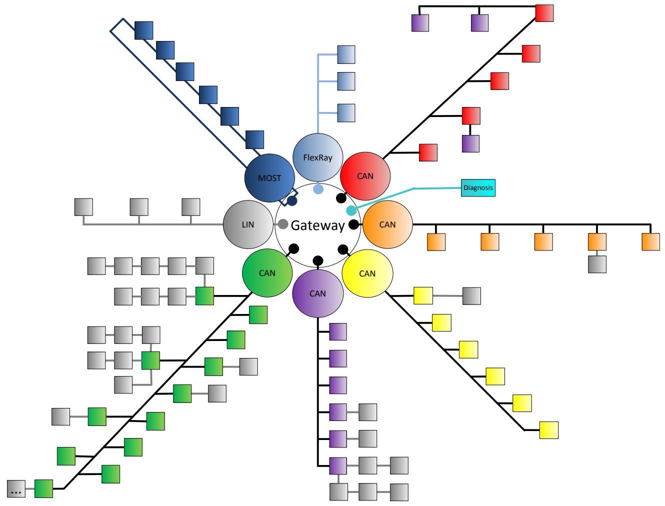


Fig. 4. The complexity of the networked control units in a contemporary vehicle is very high. Here the network of a vehicle is shown schematically [9]. Different bus systems connect many control units and share information via a gateway. In the vehicle a diagnosis interface can provide information about errors.

In the future the networking of those systems will even get more complicated and integrated. Therefore the effort and intricacy will increase vastly. This leads to the need for efficient methods and a test strategy. The system test is the last test instance before vehicle testing. Here the software, hardware and peripherals interact in a whole system for the first time. The main reasons for the system test are [19]:

- In the previous test instances such as hardware or software integration or module tests the base of the tests are technical specifications relying on the perspective of the software manufacturer. In the system test the system is seen from the perspective of the customer. The testers validate if the requirements are realized fully and adequately.
- Many functions and system features result from the combination of many or all system components and are only observable and testable on the level of the overall system.

The testing groundwork can be all documents or information that describe the test object on system level. The aims of the system test are the following, [19] and [18]:

- Errors and not documented or forgotten requirements shall be discovered before economic and physical damages arise. The residual failure probability shall be as low as possible, but the exact number will always remain unknown.
- It shall be validated if and how good the system fulfills the functional and non-functional requirements.
- Reliance on the product shall be built up. This aim is reached by testing as many functions as possible without finding many errors. Since the system test is executed several times due to changes and updates the reliance should rise with every test cycle.

Of course the system test needs a special tool environment that is as similar to the later product environment as possible. In case of the ACU this means all bus systems that simulate other control units, peripherals like sensors and physical surroundings have to be given. One approach for such a test system environment will be shown in the next chapter.

III. Research objective

For efficient and reliable system testing in a functional safety project it is important to use a test strategy that is depending on the assessed requirements for the system and that defines the test activities. Figure 5 shows one possible approach for the combination of a test strategy and test activities. After the requirements are defined in documents and tools the safety concept is derived from them. Depending on it a test strategy shall be defined using different test methods with differing priorities. The most important column have to be test cases that are systematically identified. Here all test methods of Figure 1 are highly recommended since they are part of ISO 26262. Also classification tree methods, equivalence classes, boundary value analysis and control flow analysis can be methods for finding test cases. Systematically identified test cases are proposed as most important ones since the error detection is the highest here. By proceeding systematically a sufficient coverage of requirements, functions and architecture can be reached. Those methods are not only recommended but crucial for the system test. Those methods are therefore already used extensively already.

The second suggested column are maneuver-based test cases. Since later on in the development cycle a large number of expensive and longsome vehicle tests are done it is desirable to relocate and diminish the effort as far as possible. Often vehicle driving tests are imitated by injection signal traces into the system that were recorded in a vehicle. Unfortunately the flexibility of this approach is very poor since only a small number of driving situations is being recorded. An innovative method to increase the flexibility enormously is maneuver-based testing. This means that test scenarios for the test object are defined and simulated in a suiting environment. In our specific case a vehicle dynamics and environment simulation is used to create the necessary signals for the airbag control unit. This is done since the safety algorithms depend mostly on sensor signals retrieved from vehicle dynamics and environment conditions detected by e. g. radar or camera systems. An example for a critical scenario could be breaking and skidding on ice. The parameters of this scenario such as vehicle velocity, steering angles or brake pressure can be varied and therefore an enormous diversity of maneuvers arises which could never be reached by real vehicle test drives. By combining this vehicle dynamics and environment simulation with crash data feeding as suggested in [15] it is even possible to simulate the whole

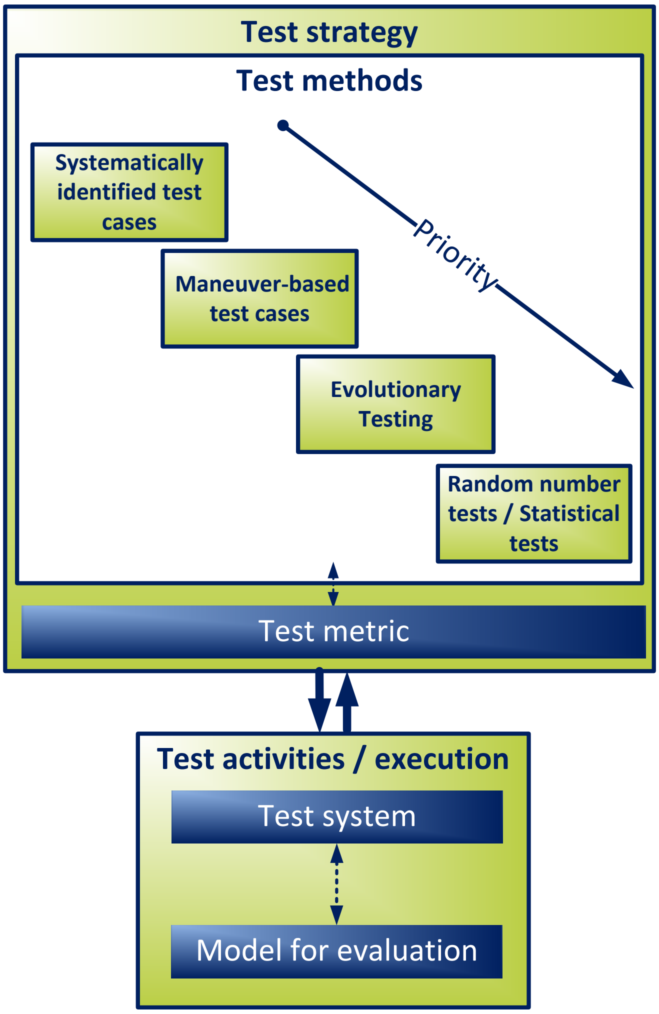

Test strategy
Test methods
Systematically identified test cases
Maneuver-based test cases
Evolutionary Testing
Random number tests / Statistical tests
Priority
Test metric
Test activities / execution
Test system
Model for evaluation

Fig. 5. Testing in a project shall rely on a test strategy that is depending on the requirements for the system and defines the test activities. The chosen test methods will always mostly rely on systematically identified test cases. A newer approach in case of airbag control units is the use of maneuver-based test cases. Evolutionary testing can either be combined with the maneuvers or used stand-alone. There is also a potential to increase error detection by random number and statistical tests as additional efficiency increasing methods. A test metric can be used to give an assertion of the progress and quality of the testing. All those tests have to be executed on a flexible and mighty test system and evaluated based on knowledge of the system behaviour and a model for localizing errors in the system.

course of an accident scenario. From normal driving situations to the precrash and incrash phases all the bus systems and signal interfaces can be fed with plausible data in order to simulate an accident. Since vehicle test drives and crash tests are done separately this simulation approach is even more realistic than the real driven tests since the whole course of events of an accident is represented. Maneuver-based testing is an innovative method for testing scenarios, algorithms and requirements flexibly, cost-effectively and efficiently. The effectiveness was very promising in various executed test measures.

The third column that is suggested is evolutionary testing as an innovative and efficiency increasing approach. A short explanation of evolutionary testing is given in Figure 6. It is still rarely used which is on the one hand due to the long time such a test takes. To create a good fitted result a test case has to be executed several dozen times over a couple of generations. This means that depending on the runtime of the single test case this can take up to hours or even days. On the other hand evolutionary tests need a lot of knowledge of the parameter or characteristic that is supposed to be optimized. Also the initial parameters should be chosen reasonably. Therefore this test method is nowadays not used for system tests, but show a lot of potential to be an additional test method for finding hidden errors that are not detected with systematic methods. Especially in combination with the maneuver-based testing it demonstrated its great potential. By evolutionary optimizing test simulation parameters especially critical test maneuvers can be created. For example the maximization of the lateral acceleration of the before mentioned breaking and skidding on ice could be a desirable characteristic of a test case. Finding such a critical test case is difficult by guessing or try-and-error. The evolutionary algorithms find the test parameters for the scenario easily and dependably. Also in case of scenarios where marginal variations of signals show different results it showed its strength. For example around thresholds where the airbag would not fire when it’s not overstepped and it fires when passed a variety of parameter sets can be created easily in order to test the thresholds properly.

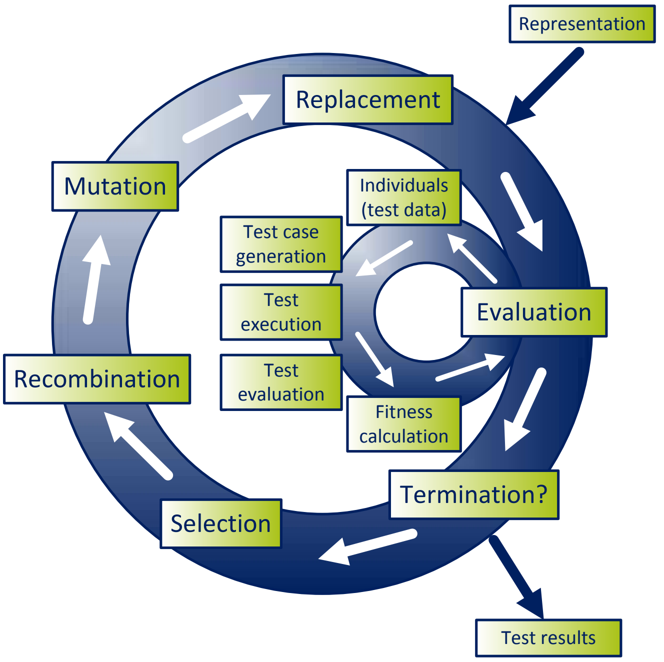


Fig. 6. In evolutionary testing test cases are identified and evaluated analog to the pattern of recombination and mutation of genes in biology. An initial population is changed and optimized until a certain desired characteristic is accomplished, [16] and [6].

The fourth column could be random number and statistical tests. Those methods are almost not used for system tests of control units respective embedded systems. They base on an extension of random number tests and use a model of the real usage of the testing object which is fed with random numbers [21]. Statistical tests are the only test class that can quantify the software reliability. Here also a great potential lies ahead to optimize test strategies. For example Monte Carlo test methods (see for example [4] or [2]) use repeated executions of random number feeding to calculate very complex systems. Of course not only completely random numbers can be used, but also pseudo-random numbers which are reproducible and random numbers within boundaries can be employed. So you can find test cases that are not found systematically or that are not considered.

All those test cases shall be considered in a test metric in order to measure the test progress [18]. Here a combination of requirements, architecture and error coverage shall be developed in order to calculate a level of confidence for the system. This metric shall give the testers an assertion about the quality and advancement of the testing activities.

In [15] an efficient and highly flexible test system for MiL, SiL and HiL tests for the airbag control unit was presented which combines vehicle dynamics simulation with crash data feeding. It is able to simulate all connected peripherals and uses evolutionary testing. With this test system shown in Figure 7 all possible errors that can occur and affect the control unit can be simulated and therefore all tests can be operated. This test system shall be the execution groundwork for the test methods. It uses a combination of the tools Messina® [1] that is used for evolutionary testing and CarMaker® [8] as vehicle dynamics and environment simulation. Both tools can also be used as application software for the MiL, SiL and HiL testing using various data bases. This enables the test system to feed the ACU with simulated and real data which can be manipulated to produce errors.

Of course finally the tests need to be evaluated. Often this has to be done by experts of the functionality manually since the creation of a test evaluation would take as long as the manual activity and would only be additional effort without benefit. But most of the test cases can be evaluated automatically and therefore a lot of time and energy can be saved. In order to do this an easy to handle measure has to be integrated into the test system. To localize at which point of the system an error occurred a model could be used. Of course it is also possible to use model-based approaches right away to derive test cases, but this shall not be the case here. The model shall be a superficial model that is only used for test evaluation. One possible approach are Markov Chains. Starting from an error free state it is possible to define error states like in a state machine. When one of those error states is reached transition rates can be assigned. This allows to collect information about the frequency of a error and the localization in the system. Another modelling approach could be Petri Nets (see [14], [12] or [3]). Petri Nets are used as a modelling measure to represent parallel processes vividly. They could be used to display the current state the system is in to show possible errors.

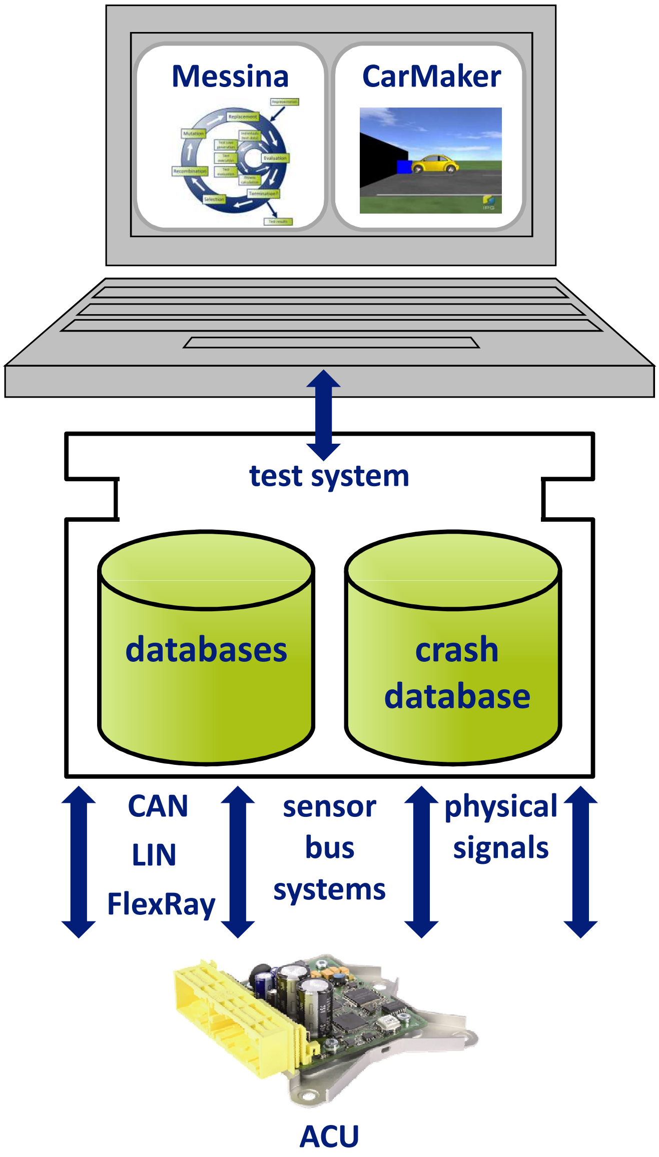


Fig. 7. The test system uses a combination of the tools Messina® [1] and CarMaker® [8]. Messina® is used for evolutionary testing and CarMaker® for vehicle dynamics simulation. Both tools can also be used as application software for the MiL, SiL and HiL testing. Using various data bases the ACU can be fed with simulated and real data which can be manipulated to produce errors.

Overall this suggested combination of test methods is one approach for an innovative test strategy for highly networked safety systems like the ACU. Of course other solutions are possible and for some methods the efficiency and feasibility in this context has to be proven. Using a variety of test methods in an efficient test strategy leads to a number of advantages [13]:

- Summarized representation of all chosen test methods
- Summarized motivation for the chosen test methods
- Consistent test principles for all phases of development
- Consistent procedure for repeating test methods
- Consistent templates, documents and reviews
- Definition of all management activities for testing
- Possible prioritization of test activities
- Describing the coherency of safety diagnoses and test

Thus the research on improving the testing for highly networked control units bears great potential.

IV. Example and Evaluation

In order to show the benefits of the combination of the methods an example configuration is shown in Figure 8. The figure shows example test sets of the different methods and explains the benefit of the usage. Also the recommended extent of the methods for the system test is given. The single test methods should be highly linked with each other in order to gain synergies. For example maneuver-based and evolutionary tests can be combined in order to find especially critical scenarios. It is also possible to test requirements of the first column by a maneuver-based test execution. Random number tests could be done in a range of equivalence classes, so that column one and four are combined. Altogether the chosen set of methods is very promising since a good test coverage can be reached by systematics and the control unit is tested in its later environment and use scenarios by maneuver-based testing. Furthermore evolutionary testing leads to a better choice of critical maneuvers and marginal situations. Random number and statistical tests can help with finding hidden errors and deepening the test coverage by combining single errors.

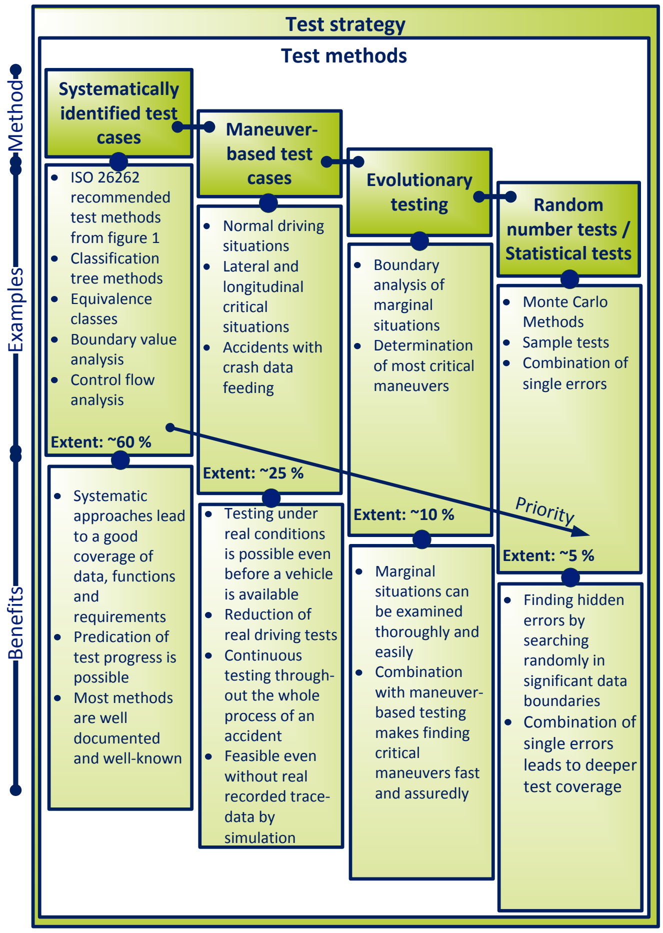

Test strategy
Test methods
Method
Examples
Benefits
Systematically identified test cases
ISO 26262 recommended test methods from figure 1
Classification tree methods
Equivalence classes
Boundary value analysis
Control flow analysis
Extent: ~60 %
Systematic approaches lead to a good coverage of data, functions and requirements
Predication of test progress is possible
Most methods are well documented and well-known
Maneuver-based test cases
Normal driving situations
Lateral and longitudinal critical situations
Accidents with crash data feeding
Extent: ~25 %
Testing under real conditions is possible even before a vehicle is available
Reduction of real driving tests
Continuous testing throughout the whole process of an accident
Feasible even without real recorded trace-data by simulation
Evolutionary testing
Boundary analysis of marginal situations
Determination of most critical maneuvers
Extent: ~10 %
Marginal situations can be examined thoroughly and easily
Combination with maneuver-based testing makes finding critical maneuvers fast and assuredly
Random number tests / Statistical tests
Monte Carlo Methods
Sample tests
Combination of single errors
Extent: ~5 %
Finding hidden errors by searching randomly in significant data boundaries
Combination of single errors leads to deeper test coverage
Priority

Fig. 8. Here an example configuration of test methods and their benefits are pictured. The test methods are highly linked. For example maneuver-based and evolutionary tests can be combined in order to find especially critical scenarios. It is also possible to test requirements of the first column by a maneuver-based test execution. Random number tests could be done in a range of equivalence classes, so that column one and four are combined. This leads to further synergies between the chosen test methods.

V. Conclusion and Future Work

This paper shows an approach to a combination of test methods and wants to be the groundwork for the development of an efficient test strategy for the system test of networked safety systems. The objective is the research what potential different existing and innovative test methods imply and how the system test can be optimized. The groundwork for this research is the system test of airbag control units. The first step towards an efficient concept of test methods was an analysis of the current testing activities in the development of airbag control units. As next step the main problems and difficulties in contemporary testing have been identified. With those data the different test methods partly already been applied and executed. This means that systematic identified, maneuver-based and evolutionary tests have already been developed in prior work and evaluated. Random number and statistical test methods and the test metric have to be given a trial in order to evaluate their possible benefit for error exposure. For the execution of all those test cases the test system still has to be enhanced and optimized. The evaluation will be done with a model of the system which still has to be created and simulated. In the end there will be a first realization and validation in order to further proof the benefits. With this information it is possible to examine the efficiency of the different methods and the potential positive effects on finding hidden errors. Resulting of this research an enduring test strategy shall be derived and documented.

Acknowledgment

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

This research is being funded by the Federal Ministry of Economics and Technology (Grant Number KF2122306PR1) based on a decision of the German Bundestag.
We would like to thank our partners IPG Automotive GmbH in Karlsruhe and Continental Automotive GmbH in Regensburg and our former partners Berner & Mattner Systemtechnik GmbH in Munich and Otto-von-Guericke University Magdeburg.

References

[1] Berner & Mattner Systemtechnik GmbH, *Messina® 3.3*, Munich, 2012.

[2] K. Binder and D. W. Heermann, *Monte Carlo Simulation in Statistical Physics*, Springer Verlag, Berlin, 2002.

[3] C. G. Cassandras and S. Lafortune, *Ereignisdiskrete Systeme*, Springer Science+Business Media, New York, 2008.

[4] G. S. Fishman, *Monte Carlo*, Springer Verlag, New York, 1996.

[5] *FlexRay, The communications system for advanced automotive control applications*, http://www.flexray.com/, February 2012.

[6] I. Gerdes, F. Klawonn and R. Kruse, *Evolutionäre Algorithmen*, Vieweg Verlag, Wiesbaden, 2004.

[7] International Organization for Standardization, *ISO 26262: Road vehicles - Functional safety*, November 2011.

[8] IPG Automotive GmbH, *CarMaker® 4.0*, Karlsruhe, 2012.

[9] T. Kiesewetter, *Der VW Touareg - Karosserieelektronik und Infotainment*, ATZextra, No. 2010-02, page 47, 2010.

[10] W. Lawrenz and P. Bagschick, *CAN - controller area network*, Hüthig Verlag, Heidelberg, 1999.

[11] *LIN, Local Interconnect Network*, http://www.lin-subbus.org/, February 2012.

[12] J. Lunze, *Ereignisdiskrete Systeme*, Oldenbourg Verlag, München, 2006.

[13] H. Paulus and F. Eimbeck, *Testen in Projekten mit Funktionaler Sicherheit*, Hanser Automotive electronics systems, November 2011.

[14] C. A. Petri, *Kommunikation mit Automaten*, Schriften des Rheinisch-Westfälischen Institutes für instrumentelle Mathematik der Universität Bonn, Bonn, 1962.

[15] A. Raith.,K. Sattler, R. Ertlmeier, and T. Brandmeier, *Networking and Integration of Active and Passive Safety Systems*, Ninth Workshop on Intelligent Solutions in Embedded Systems IEEE WISES 2011, Conference Proceedings, pp. 75 - 80, 2011 University of Applied Sciences Regensburg, July 2008.

[16] J. Reiner and J. Meyer, *Evolutionäres Testen von Steuergeräten*, Elektronik automotive, September 2010.

[17] A. Siller and D. Korotkiy, *Systematisch zu Testfällen*, Hanser Automotive electronics systems, November 2011.

[18] H. M. Sneed, M. Baumgartner and R. Seidel, *Der Systemtest*, Carl Hanser Verlag, München, 2012.

[19] A. Spillner and T. Linz, *Basiswissen Softwaretest*, dpunkt.verlag, Heidelberg, 2010.

[20] W. J. Stewart, *Probability, Markov Chains, Queues, and Simulation*, University Press, Princeton, 2009.

[21] J. Wegener, *Evolutionärer Test des Zeitverhaltens von Realzeit-Systemen*, Shaker Verlag, Aachen, 2001.