KISS – A universal approach to the development and design of occupant restraint systems

Dipl. Ing. André Neubohn, Volkswagen AG
Dipl. Ing. Christian Weiss, Audi AG
Dr. Frank Keck, Audi AG
Dr. Andreas Kuhn, Andata Entwicklungstechnologie GmbH & Co KG

ESV-Conference
Washington DC, June 6-9th, 2005

Abstract

Current passenger cars offer an assemblage of complex systems for the protection of occupants in different accident configurations. The adaptivity of the systems will widen in the future, i. e. the systems will be adapted to offer optimized and increased protection for different occupant classes during serious crash situations. This will lead to an augmentation of system complexity. Only through an intensive application of CAE based methods is one able to a) chose the appropriate system components and b) assess and optimize the interaction of the latter to fulfill the requirements.

The competence of developing and assessing new features is one of the core tasks of car manufacturers. To satisfy this demand, Audi and Volkswagen started the KISS (**k**ey competence **i**ntegrative **s**afety **s**ystems) project. The main goal of KISS is to increase the development and assessment competence of occupant restraint systems throughout the complete development process, which consists of the actual vehicle, the occupants, the restraint systems, sensors, airbag control unit and the algorithm which is implemented to control the deployment of protective measures.

Because KISS kicks in at the very beginning of the development process when essential properties are yet to be defined and boundary conditions are still fluid (e. g. package, system architecture,...), KISS is able to lay the groundwork for an effective and – concerning its complexity – controllable occupant restraint system. Along with conventional car and occupant simulations FEM crash simulations can also be increasingly used for the optimized placement of crash sensors and the computation of sensor signals. Using modern mathematical methods of signal classification, these signals are utilized to generate a first implementation of a crash classifying algorithm. Using stochastical and statistical methods the robustness of a solution can be assessed in a qualified way long before hardware for tests is actually available.

The universal and integrative design of the system is driven by the requirements. Starting with the global request “Protect each occupant as well as possible in each crash situation” one can derive different requirements for the restraint systems, the control unit, the sensor system and the crash algorithm. KISS enables considerable acceleration of the entire development process.

The realisation of new, innovative systems is only possible in close collaboration with system suppliers. A structured approach based e. g. on the V-model starts with a detailed analysis of the requirements to be satisfied. Based on these requirements different solution concepts are created. One of the concepts is finally chosen and implemented. Despite the different focus of car manufacturers and system suppliers in the development process, it is crucial to build up overlapping areas of expertise and competence to jointly develop innovative, robust and cost-optimized solutions.

The following presentation gives a survey of the content, the interaction of the processes and technologies used in the KISS project and their impact on the future role allocation between OEM and system suppliers.

Introduction & approach

Today’s vehicle safety systems are characterised by high levels of functionality and more and more demanding product requirements. In some cases, this leads to very complex systems. For this reason, it is difficult to adapt existing systems to new requirements or to guarantee a sufficient degree of quality from the outset of a new development.
When using traditional, hierarchical and purely deterministic developmental methods, the necessary development requirements are inordinately dependent upon:

- the number of requirements
- the number of functions
- the degree of complexity

This is especially true of vehicle safety. Over the last few years, vehicle functional requirements have increased significantly, both in scope and complexity. And there is no end to this rapid development in sight. On the contrary, it is anticipated that the number of requirements will rise significantly. Passive safety is a particularly good example that can be used to illustrate how functional requirements have increased over the last few years (Figure 1).
A paradigm shift in the approach to product development appears inevitable, without which the dramatic increase in development costs and resources (development effort) cannot be countered.

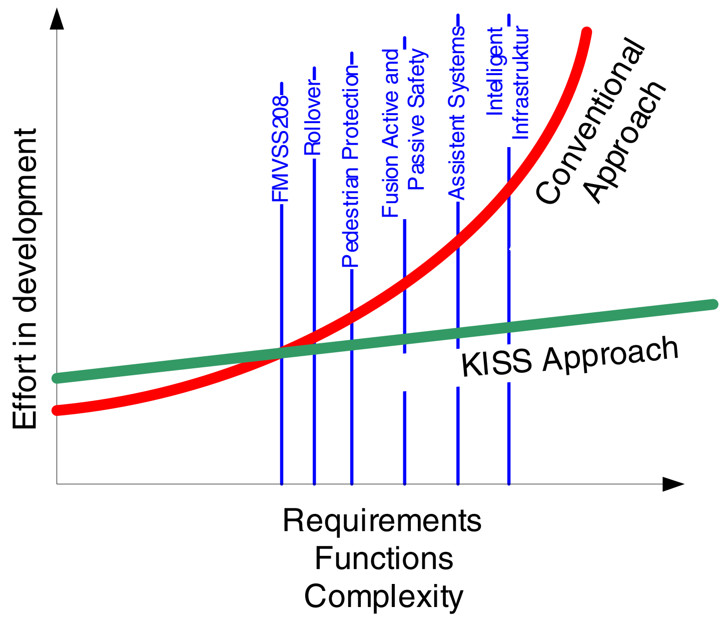

Figure 1: Development effort vs. increasing requirements, functions and complexity.

The solution

As part of the scope of the KISS (**K**ey competence **I**ntegrative **S**afety **S**ystems) project, new technologies are to be introduced to the development process in order to check the disproportionate increase in development effort and secure control over system complexity. This will take place by way of:

- an increased use of process models within the development process for mechanical components and control software, with a view to creating a targeted and requirement-driven approach,
- the application of modern mathematical modelling techniques from the field of multi-variant data analysis and soft computing as a means of containing and maintaining control over complexity, and
- the application of stochastic processes for robustness management and for the handling of uncertain data.
- the rethink of the allocation and understanding of roles between OEM and suppliers.

No one person alone is able to make the breakthrough suggested by the above. Only a targeted, requirement-driven and suitably integrated implementation can produce the desired effect.

Organisation structure and role allocation

When looking at the occupant restraint system from a systematic, global perspective "Intelligent control" should be considered as a separate component. It is of crucial importance since it drives the functional integration of various hardware components (sensors and actuators). This is what we would call a key technology, and the essential element needed to meet current and future requirements.
A logical consequence of this is that the OEM takes on this task as a core competence of global system integrator.
Only the OEM is in the position to implement "intelligent control" calculations and associated mathematical evaluation models within the early concept-led development phase, and thus do justice to its integrative nature. Future requirements cannot be fulfilled efficiently without the active, constructive integration of algorithms and mathematical control models within overall system development. Thus, competence in control logic is a decisive factor in being able to maintain control over the increasingly complex systems of the future.

This need will be intensified by the increasing heterogeneity of electronic hardware and further separation of hardware and software. Open databus systems and increased integration of all different kinds of control systems (active and passive safety) will mean that integrative capabilities will increase in importance.
In particular, topic areas such as sensor fusion, which has been the subject of intense discussion, will not be able realistically to be implemented without such core competences. On the other hand, this will not mean a decline in the importance of the system supplier. Instead, a new kind of partnership is required, characterised by a greater intensity and improved quality. The OEM shall supply concept proposals and requirements to the system supplier in a much more professional and homogenous form, thus improving the basis for fine-tuning and final implementation.

Process models and requirement management

A process-based control of disparate functions and high levels of complexity require new approaches to the development process, which will be driven by software technology.

Over the last years, formal process models have become more established. There are two reasons why. Firstly, because of a need to improve Quality Assurance measures in the development process. Secondly, through the use of synergies - the extensive reuse of similar process models in development processes. Examples of these include the *object-oriented development methods* or the *V-model.*

These process models have a number of variants, which, aside from their detailed attributes, have more or less the same basic structure: first of all, an intensive **requirement analysis** is carried out and a requirement model is created. Based on this model, one or more concepts for the **system blueprint** are developed. The chosen concept is then constructed in a **system development phase** and fed into the concrete **implementation** process.

It is important here that the process begins with a thorough requirements analysis which will then be used to drive development. This may sound like common sense but it is not always applied in practice. In fact, often the exact opposite is true:
For existing systems, new requirements are only implemented incrementally without having carried out a proper requirements analysis and without sufficient examination of the effect on the global requirements complex. Development is driven by the solution which is already in place, avoiding all but the most essential changes. Many innovations stand alone from the global system, on the level of individual components. This means that system development is component-driven, in other words by a detailed technical solution, often without sufficient assessment being carried out of its impact and weighting within the system as a whole.
The above procedures are pragmatic, you may say feasible for simple, well-known systems. When complex systems are involved, the cumulative effect of looking at requirements on an incremental basis is often underestimated or not considered at all. There is then the danger of finding oneself at the end of an ill-conceived trial and error scenario.

Today, where a requirements analysis is carried out at all, it often consists of nothing more than a simplistic, more or less structured collection of requirements in a database or Excel file. Occasionally, a specialised tool (such as DOORS) might be used to collate requirements.
Object-oriented software development technology takes this a stage further by modelling the requirements with UML for example, and generating code (proposals for technical realisation) directly from the requirement models. This enables requirement conflicts to be identified earlier.

A similar requirements analysis can also be carried out for mechatronic systems. Requirement conflicts can be detected and resolved very early on in the development process. This can be achieved by representing the functional requirements of a mechatronic control system's classification module in mathematical form, using concrete examples from the planned behaviour. Multivariate statistical methods, such as cluster analyses, can be used to reveal conflicts between required firing times for certain sensors for example. This also works when explicitly taking into account tolerances and uncertainties.
Requirement management becomes more than filling out checklists. Instead, it is a constructive, engineer-supported development tool that is far more than just an approval criteria applied retrospectively.

Mathematical modelling

The control algorithms are the essence of all "intelligent systems" and are described using mathematic models. By using a suitable mathematical formulation for each requirement, statistical regression can be used to derive the mathematical model for the implementation of the control task: with the proviso that any requirement conflicts are resolved first.

This procedure is universally applicable, and not just limited to the current crash classification. The creation of mathematical models to describe systems cannot only be applied directly to control systems, but also aids the decision-making process during development. It will bring about a similar wave of innovation, as numerical simulation for virtual product development once did in the design of structures.

The required mathematical basis has been fully developed and is readily available. Methods such as

- Multivariate data analysis,
- Statistical regression and prediction models,
- Fuzzy logic,
- Neural networks,
- Machine learning,
- Stochastic validation

will be used on a more broader basis in the future as development tools, and not only by specialists.

The way tasks are created for signal classification can be applied universally, and is also applicable to many other applications. A general classification framework in the middleware of embedded systems will no longer just be a dream. Today, a classification object library could be created allowing the processing of sensor signals to be standardised for simple sensors. The functionality would be then provided by parameterisation alone, while preserving full-scale individualisation of vehicles.

Example of application

The following concrete example is used to illustrate the KISS approach. At its core are the requirements, which drive development. Figure 2 shows the basic structure of the requirements of a restraint system with the scope of passive vehicle safety.

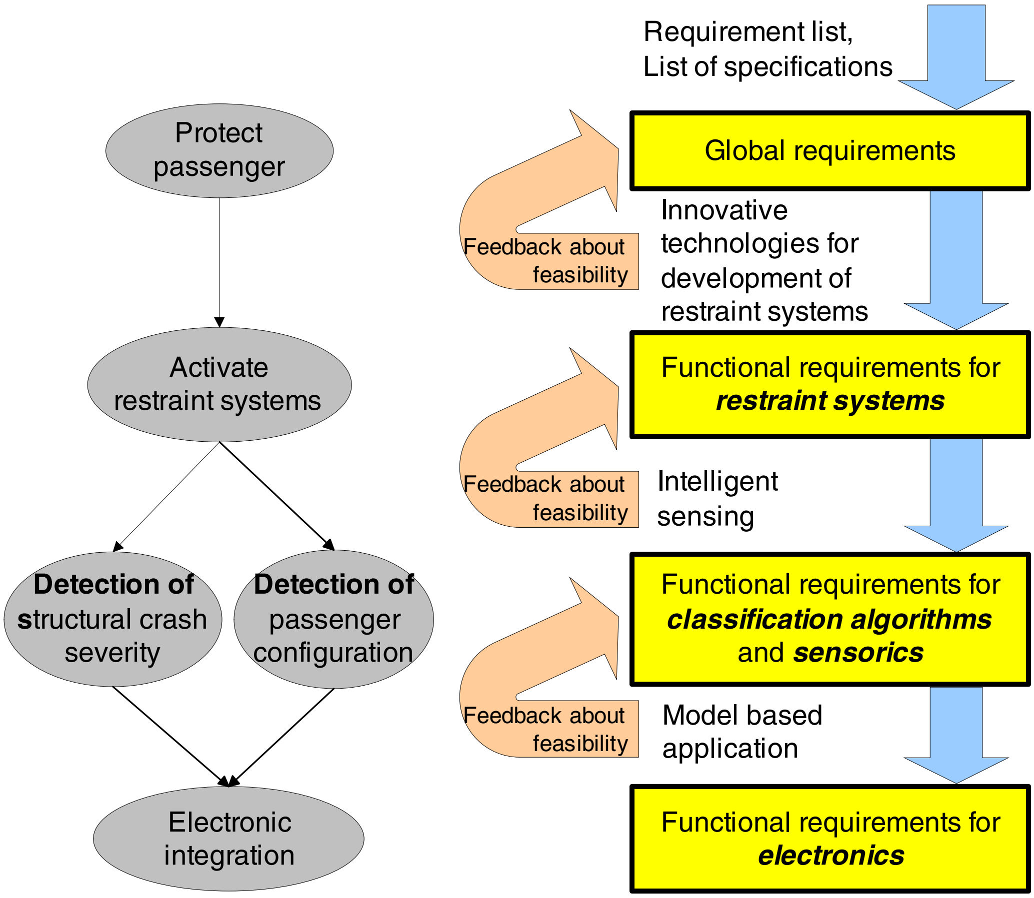

Figure 2: Requirement structure for passive safety

The upper requirement level represents the **general, global requirements**. These determine which crash scenarios vehicle occupants must withstand without serious injury, which legal standards must be met and which market requirements must be taken into account, etc.
Out of these come the technical concepts for the restraint systems: **the functional requirements,** e.g. an adaptive airbag is required to meet certain global requirements, etc.
This must then drive the selected components of the restraint system. Next come the **requirements of the control system,** e.g. that the airbags must be fired at certain, pre-defined times when a crash occurs. In turn, this requires appropriate control electronics to be installed which enable real-time realisation of these classification requirements. Thus, the firing time is a functional requirement of the control algorithm, which follows from technical realisation via a specific airbag!

In KISS up to now, the second layer used to set up the detection algorithms and sensor systems was revised on the basis of the functional requirements of the restraint system. The functional requirements of the restraint systems, such as the requisite trigger times for actuators on the basis of pre-defined sensor signals were represented using examples (crash tests and/or simulations). Using multivariate data analysis, requirement conflicts for firing times can be detected quickly and, above all, on a statistically quantifiable basis, before being fed back into restraint system design. Appropriate

statistical regression methods are then used to derive the mathematic models for signal classification from the models of requirements for restraint system control.
The objective here is to create a universally-applicable framework for a classification algorithm, in which the quasi-automatically generated classification modules can be integrated seamlessly.

In principle, the same methodology can also apply to the level of functional requirements of the restraint system. The lowest requirement level will be represented in future using model-based applications or through the availability of appropriate control unit middleware. In the future, excessive specialised knowledge of embedded programming will no longer be a prerequisite for the application of intelligent control algorithms.

At present, the requirement structure shown in Figure 2 is a reflection of the organisational structure of the car manufacturer. Each requirement level is usually handled by a separate organisational unit. As a result, a great deal of time passes before a response can be received concerning technical feasibility after an additional global requirement has been implemented. This is because all requirement levels must be considered in turn. The new technology in KISS, coupled with a reduction in response times on a sensor systems requirements level, opens up an opportunity to establish a vertical, project-based organisational structure. In the future, this structure will enable the processing of all requirement levels within a very small time frame. Based on this, qualified concept assessments can be submitted.

Integration into the V-Model

At present, the V-model is the standard method used for the development of embedded software in cars. Accordingly, the development steps described above must be adapted to this standard. Figure 3 illustrates the V-model as applied to KISS.

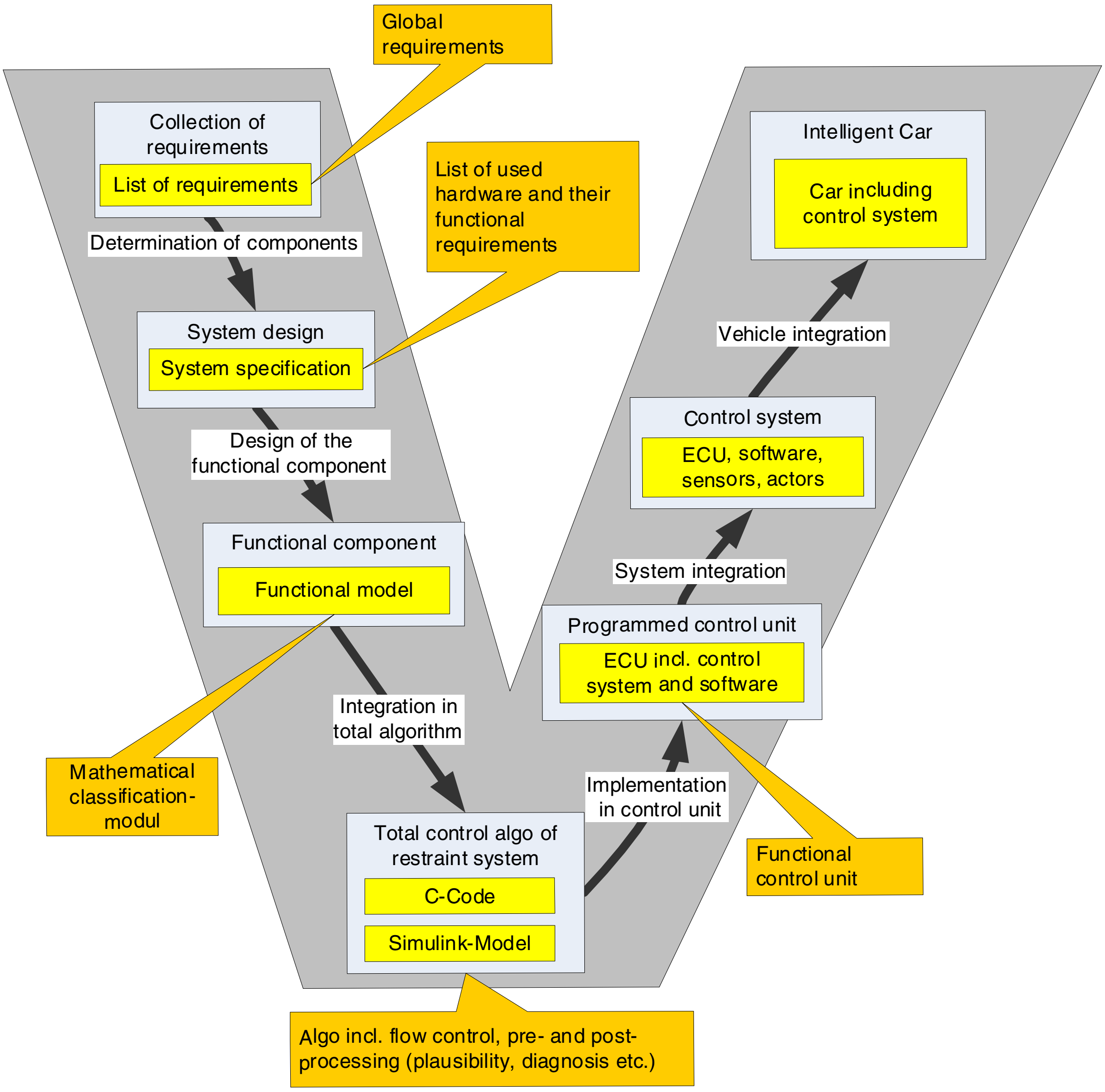

Figure 3: V-Model

First of all, a **system analysis** is carried out in the form of a **requirements analysis** with the resultant **system design** (=concept). This may be used to determine the requirements (using examples of load cases) which actually need to be fulfilled, followed by the number and type of restraint devices to be fitted to the vehicle. This is normally carried out by the appropriate technical department for the development of restraint systems. The result is a list of functional requirements (airbag firing times)

which are represented as concrete examples (test and/or simulation data) depending on the project phase.
Next, the system requirements are converted into the **functional components,** which are used to control the individual actuators. The result here is a mathematical classification module that is generated on a quasi-automated basis, e.g. to control the side airbag deployment or the second level of an frontal airbag. The technical department for restraint systems may also be in charge here. This department applies the methodology to continuously verify the functional feasibility of its firing requirements. The technical department for vehicle structural design can also use this methodology to verify sensor settings. The final implementation model for the functional components can then be fine-tuned by the technical department for algorithm and control logic, for example. This department integrates the functional components within a **global algorithm for the control of restraint systems** and adapts the individual functional components to their overall effect. The global algorithm is created in standardized form as an algorithm model which is independent of final hardware implementation. The validated global algorithm model serves as a reference and template for the **implementation within the control unit**. Only at this stage is the actual hardware to be used to be considered in greater detail. The system supplier adapts the referential algorithm to his hardware and optimises runtime and memory space requirements accordingly. Correct implementation can be verified using implementation tests with specified input and output behaviours for pre-defined test cases. The system supplier is responsible for integrating the control unit into the overall electronic infrastructure with the associated sensors and actuators. At the end of the process, the vehicle manufacturer will approve and integrate this into the vehicle.

These days, much of the time and effort is concentrated on the right-hand branch of the V-model. A reason for this lies in the technical status of embedded systems, for which software must be programmed in very close proximity to hardware requirements. This means a large adaptation effort is often required even for minimal changes. These days, electronic details play any extremely important role, and in some cases can even influence the system concept. In future, the databus used for sensor communication must not be important for a restraint system. The architecture of the trigger logic should be almost entirely independent of such an electronic configuration.
In future, software and hardware will be to a large extent separable from each other. The operating systems and middleware required here (programming language for embedded systems) are already in development and will be available in the next few years. The right-hand branch of the V-model can for the most part be implemented automatically. This releases the energy required to address the left-hand branch more intensively. This is essential for further development since for the most part the quality of the whole system is already determined in the first two blocks of the V-model. At present, these blocks are given insufficient attention on account of the predominately solution-based and requirement-driven approach currently in existence.

Validation and safeguard strategies

In the future, validation and system safeguarding will assume a more important role than at present.

In the main, this applies to the functional level of mathematical system models used to assure the functionality of control and classification modules. Considering the number and great variety of requirements this task is by no means an easy one. Increasing functionality means it is almost impossible to thoroughly test all combinations in the time available, let alone within a hardware test (Each discrete new function compounds the number of possible system configurations). Stochastic validation methods and probabilistic approaches will be the best tools to assure system functionality.

In addition, the aim here is also to assure the underlying design data, which in future will be principally taken from numerical simulations. In order to prevent problems from arising, these simulation data must be subjected to quality checks. The stochastic nature of the underlying problem (a crash is not deterministic) means that stochastic validation methods are also well suited here.

Formal validation will become more important due to new role models with supplemental/modified process interfaces. A high degree of validation is required to enable conceptual designs/ requirements to be transferred in qualified form.

Summary

As an element of the KISS project, a key design concept has been drawn up for the future development process of intelligent control systems. The following points can be summarized in this regard:

- “Intelligent control” has been identified as a key factor in controlling vehicle safety systems which are likely to increase in complexity in the future. As a result, a greater focus will be placed on taking this into account early on in the development process.
- It will become increasingly important to consider system development on a holistic and integrated basis.
- System modelling using modern mathematical methods represents the underlying core technology. This can be applied directly on a control unit level and also used to aid decision-making processes within development.
- The use of KISS not only means that parts of a classification algorithm are taken on by the OEM, but represents an entire paradigm shift in the whole development process. The resulting benefits come about through a targeted combination of:
    - example-based methods
    - an object-orientated approach
    - use of stochastic methods and other soft computing methods

- The organisational scope for targeted use is provided by software technology with formal approach models, such as object-orientated development methodology or the V-model.
- The development process should always be driven by requirements and not by already available technical solutions.
- A requirement-driven process structure means that development is also driven from the state brought about by the requirements. In general, this is the design of the restraint system and not the electronics.
- This methodology can be integrated seamlessly into the V-model.
- The car manufacturer can concentrate on the left branch of the V-model. In other words, on core themes cost and quality determiners.
- The right branch of the V-model is the remit of the system supplier of the electronic control systems.
- The interface between the vehicle manufacturer and system supplier can be a global standardised algorithm model, which can become the open industry standard. The differentiation on account of function is provided by the functional modules. These can automatically be generated using suitably edited requirement data based on practical examples.
- The result of intensive cooperation between the OEM and system supplier will be an improvement in quality.

Conclusions and outlook

Continuing to use the approaches within KISS will result in the following scenarios for the future:

- The methodology will be extended across the level of requirements made of the restraint system (vertical expansion). In other words, in future this methodology will also be used more intensively in the development of the restraint system itself, not only on the sensor system/algorithm level.
- The methodology will also be applied to other applications, perhaps making use of synergies within sensor evaluation (horizontal expansion and sensor fusion). An example here is the fusion of active and passive safety.
- In the foreseeable future, the right-hand branch of the V-model will become more standardized using better hardware, common operating systems, suitable middleware for embedded systems and the resultant increase in automation. The separation of hardware and software will make it possible to concentrate on the real function of control tasks, of the formulation of the “intelligence” within engineering, without being strongly limited by electronic hardware realisation.
- Generally speaking, existing control tasks are universally applicable. For this reason, the form of the classification module can become the standard, which will be applied universally in the middleware of the software of all control units.

- Driven by requirements, vertical, project-orientated organisational units can realise prototypes of intelligent control systems within a very short period of time.
- The OEM can direct most of its attention to the concept and system design phases within the development process, where quality and cost considerations are decisive.