INTEGRATION OF FUNCTIONAL SAFETY IN THE DEVELOPMENT PROCESS

The new safety standard, ISO 26262, forces automotive electronic system manufacturers to align all their development processes with safety considerations. To date, safety activities are often still undertaken quite independently from the developers' "normal" work, however, with the consequence of safety requirements entering the development process at an advanced state causing costly additional development work. A new methodology from Berner & Mattner now enables to integrate conventional systems development closely with safety activities with the help of tools, so incorporating safety into the system design right from the beginning.

AUTHORS

DR.-ING. BERNHARD KAISER
is Team Leader Automotive
Competence Center Safety &
Systems Engineering at Berner &
Mattner Systemtechnik GmbH in
Berlin (Germany).

JÜRGEN MEYER
is Manager Automotive at Berner &
Mattner Systemtechnik GmbH in
München (Germany).

REFERRED TO ISO 26 26 2

Making reference to the latest science and technology state of the art at any point of time, the Product Liability Act and the new automotive standard, ISO 26262, in combination require vehicle manufacturers to comply with any and all stipulated safety activities across the entire product lifecycle. In this context, all work results need to be accounted for consistently – from hazard analysis to the safety concept on to implementing and testing the resulting safety requirements – accompanied by a broad range of reviews, safety analyses, and assessments. To enable this substantiation in liability cases even years later, traceability, that is, the ability to draw consistent conclusions from revealed failure potentials to associated safety requirements and affected system components on to verified test cases, is needed above all. Given the complexity of today's products, this can only be mastered with the help of tools.

Today, this presents many organisations with problems for the reason that in many cases, safety activities take place independently from the remaining [BK3]product development process to a great extent. In many cases, hazard and safety analyses relate to out-dated assumptions concern-

ing the system's functionality and architecture because different staff members are often assigned with the respective tasks.

DEFICIT

Safety concepts are formed independently from the concurrently advancing system design, which results in the safety requirements entering the development process unduly late. This, in turn, results in costly redevelopment cycles to integrate the safety aspects ex post – along with the additional tests and reviews they require.

The fact that safety managers and FMEA moderators often speak different languages and use other tools than system and software developers is a main issue in this context. This leads to tool based architecture models, which are already available today in many cases, remaining unused when it comes to examining error propagation from one system component to the next in a systematic way. Conversely, safety managers fail to prepare their safety requirements in terms of language or tools in a way enabling them to be imported and linked into regular requirement specifications. So, although safety managers, FMEA moderators, and developers all do a good job for themselves, lack of continuity frustrates any quest for efficiency. The challenge of today for the organisations in question is to avoid parallel processes and to integrate safety activities into the development process from the beginning.

THE NEW WAY: HOLISTIC, HIERARCHICAL, MODEL-BASED

Working in several projects, Berner & Mattner specialists devised a development method combining hierarchical modelling of system architectures and failure nets, allocating safety mechanisms, and systematic requirements tracking including proof of test coverage. This methodology consists of combining basically known and proven approaches in an intelligent, tool-integrated way. As such, the methodology represents a procedure instruction with no compulsory reference to any specific tool; it has, however, been employed for production development processes at several automotive component suppliers in pilot projects using standard tools such Doors, APIS IQ-FMEA und Enterprise Architect. Transferring the methodology to

other tools on the market is possible with little effort.

A further advantage of this approach lies in the fact that the resulting models and concept building blocks are modular and reusable and so can be utilised efficiently for product variants and future developments.

START OFF SYSTEM DEVELOPMENT

As in any mature development process, a carefully structured requirements specification in a tool such as, e. g., doors, marks the beginning. Working with the customer, a structure was prepared, structuring requirements, assumptions, and constraints by functional areas and defining so-called features as its bottom level. These take the form of keyword-like functional descriptions, e. g., "motor current control".

This course of action allows to establish first links of this manageable number of features to the system architecture model, which is concurrently forming in a modelling tool, at an early stage and with little effort, ❶. This implies developing a very simple model of the planned system architecture at first. It is then iteratively refined and reworked multiple times by defining subsystems and listing external signals/ inputs and including the system's dynamic behaviour.

A crucial aspect of the system modelling is its hierarchical structure, which is extended to an extra level beyond the system boundary for the purpose of subsequent safety modelling. This level takes the form of a context diagram, depicting the system in its interaction with neighbouring systems. It is important for all levels to identify the interfaces between the components in full at an early stage for from the safety experts' view, the impact of failures will propagate across these interfaces as well. For example, an abnormally low motor speed sensor value can lead to an excessive torque command in the following control unit [BK6]taking the effect of unauthorised drive acceleration at the system's external boundary, to be characterised as a safety-critical hazard.

AUTOMATED EXPORT

Owing to the equivalence relation between the hierarchical system model and and its structure tree, ❷, the latter can be

ibd [Block] ElectricalDrive [Extended]
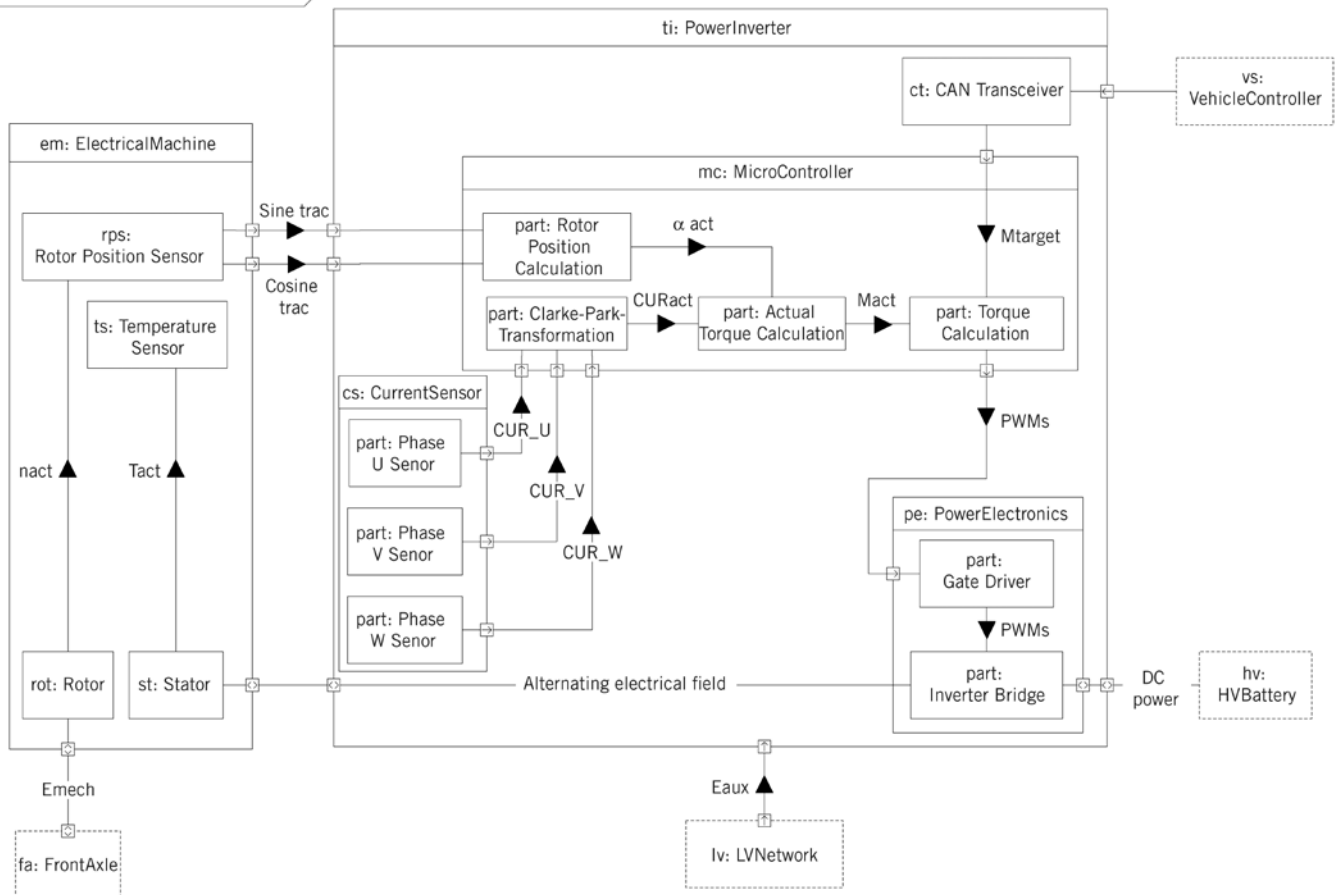
❶ System architecture represented as a SysML inner block diagram

imported from the system architecture model into a hierarchical type FMEA tool automatically with the help of scripts. This enables Safety and FMEA experts

tocbenefit from the data structure now available to them while functional developers are iteratively refining requirements and the system's architecture.

Moreover, the link-up between Doors and the modelling tool allows to incorporate the intended functionality of each component on all hierarchy levels into the

bdd
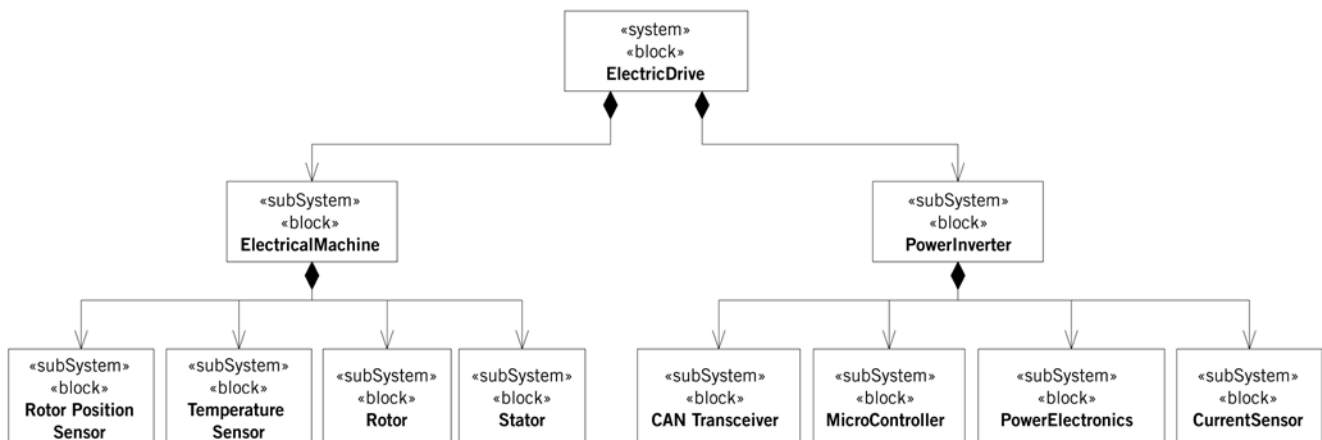
❷ Representation of the same system as a composition tree
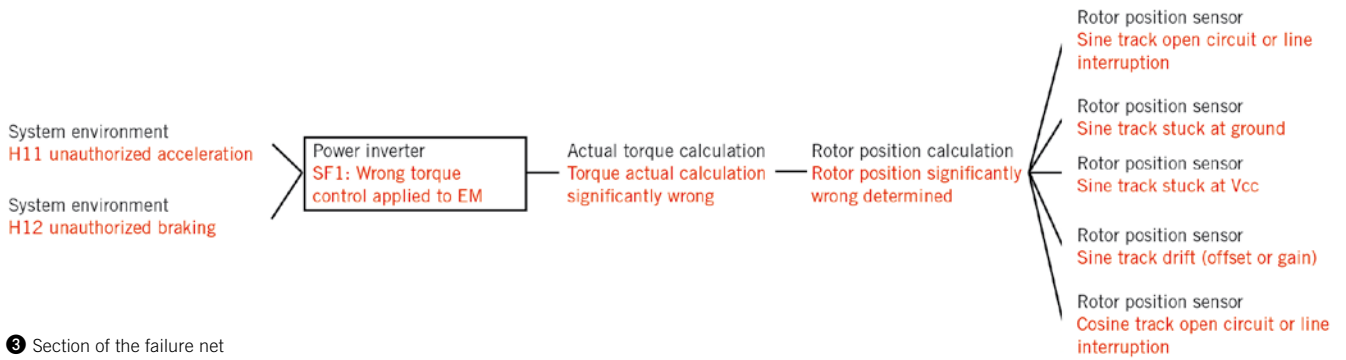
**❸** Section of the failure net

FMEA automatically as well, enabling both to reduce the effort needed considerably and to attain a level of consistency not previously known.

The required safety analyses can now be conducted using the safety tool, starting with a hazard analysis and a risk assessment. This is done by associating the intended functions with their possible malfunctions in a systemised fashion with the help of a keyword method taken from the HAZOP technique (e. g., "too high" or "too low" with continuous signals and "too late", "unexpected", etc. with events). The cause-effect relations of the components' faults are pictured graphically as failure chains

in the tool, ❸. The tool automatically provides appropriate parameters derived from the known interfaces and their interconnections defined by the architecture, the reason being that faults can only propagate between neighbouring components or between components and their superordinate components along these paths.

The topmost architecture level represents the overall system within its environment so failures taking effect at its boundaries are potential hazards. These are assessed for their "severity", "probability of exposure", and "controllability" parameters in relation to various traffic situations to obtain the ASIL classification.

So a natural linkage between the hazard analysis and FMEA already exists, searching the system design right down to the component level for newly emerging fault possibilities as it grows ever more concrete in the course of the development work. Collaborating in workshops, developers, the FMEA moderator, and safety experts then jointly define technically and economically feasible countermeasures according to the provisions of ISO 26262 serving to prevent, detect, and control possible failures. The consistency between FMEA and the system model has proven to be an excellent aid for mutual understanding between the different specialist areas.
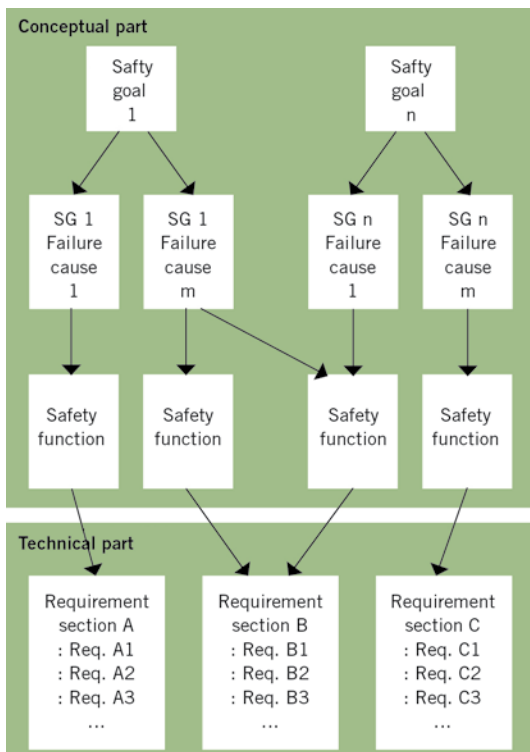
❹ Doors representation of the safety concept
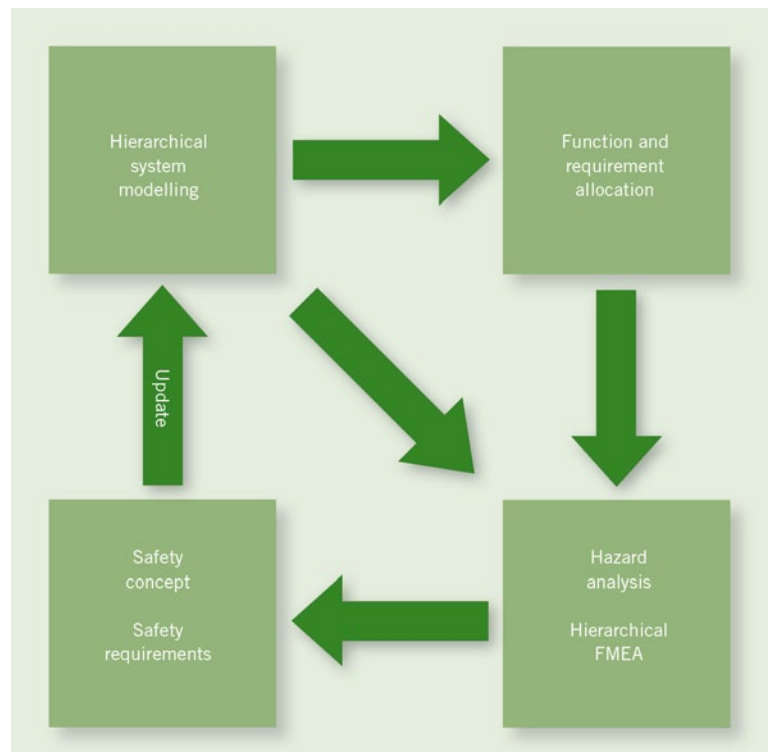
❺ Integration of the technologies into a consistent framework

FEEDBACK INTO THE DEVELOPMENT PROCESS

Detailing measures related to safety requirements and their allocation to the system architecture is done in the safety concept according to the standard's provisions. In Berner & Mattner's approach, this is not updated as a text document any more, but rather completely within Doors. Importing safety measures from the FMEA tool offers automation potentials in this direction as well. Doors attributes allow to classify statements in the safety concept, as assumptions, measures, safety strategies (such as, e. g., ASIL decomposition) or, finally, technical requirements, for instance. Only these last statements enter into the continuing requirements process and need to be covered by proof of implementation and test cases. Using a requirements tool for the safety concept bears many advantages: Apart from its hierarchical structuring capability, the tool enables the use of attributes as stipulated by ISO 26262, motivates users to employ atomic requirements, and allows to integrate them into the set of Other Requirements, including their subsequent linking to the system design and to the test cases. Due to the initial, theme-related grouping of requirements with features allocated on the model level, each system component's ASIL can readily be determined. The stipulated safety concept reviews can be simplified considerably by "clicking along"; safety measure justifications are understandable at any time, which is essential in case of subsequent revisions in particular. The baseline functionality continues to allow for the required configuration management, ❹.

Test cases were also managed in Doors, each complemented by attributes revealing the states of test case completion and test result. This allowed to generate metrics and pie charts automatically on a weekly basis showing the safety requirements coverage status and so enabling the safety assessment to be prepared in an optimal way.

CONCLUSION AND PERSPECTIVE

The described methodology has already been successfully employed in several series development projects for hybrid and electric drives. In the initial phase, a consistent understanding of development and safety experts concerning methodology and the system had first to be worked out in workshops and an increased number of consultations during the actual work; it soon turned out, however, that the initial modelling effort paid off rapidly already in the first project because all concerned collaborated increasingly smoothly. Furthermore, once prepared, model components can be reused in future developments due to their modular character.

Owing to the new approach, safety requirements found their way into the development cycle considerably earlier than in other, comparable projects and so could be accounted for with less effort and at lower cost. Safety requirements were deposited in standard tools such as Doors in the form of technical specifications enabling to test them together with the original technical specifications in a transparent manner. The safety concept can be mapped completely and transparently with all its links to the system mode, ❺.