An Appeal
Practical Standards in Autonomous Driving

The automotive industry is moving towards Level 3 and Level 4 automation. Recent experience with Level 2 system safety has shown that even systems that are considered industry standard are operating on assumptions that, when violated, lead to highly dangerous situations. Established methods like ISO26262 functional safety and extensions like SOTIF (Safety of the intended functionality) give top-level requirements for the safety of automated systems but no guidance on how to reach those requirements. Elektrobit believes that it is necessary to begin industry-wide discussions on best practices for algorithmic and architectural setups that define the state of the art for safe self-driving systems.

AUTHOR

Dr. Ing. Björn Giesler
is Leader Driver Assistance at the Elektrobit Automotive GmbH in Erlangen (Germany).

INITIAL SITUATION

Highly automated driving functions of level 3 and 4 still require a human driver to be in the vehicle, both technically and legally, but they allow the driver to divert their attention away from the driving task. That means that they have to be able to at least safely recognise whether the current situation can be handled automatically or must be dealt with by the driver. In level 4, they have to be prepared for the driver not being able to take over, so they must be prepared to deal with all situations safely. In level 5, the presence of a driver cannot be assumed any more; but already in the lower levels the automated vehicle, while under the control of automation, can be considered operator-less.

Functional safety, as proposed by ISO26262, rests on the basic assumption that any hazard can be described in categories of exposure (how likely is a hazard to arise), severity (how dangerous is the hazard to life and health if it arises), and controllability (to what degree can the human operator be expected to be able to react to the hazard, and avert the danger). This latter category is problematic for automated systems. Since the operator, even if physically present, has not paid attention to the scene, they can likely not be expected to handle the situation at all. Therefore, there is some merit to the question whether ISO26262 is applicable to operator-less systems. At the very least, under the control of automation, all hazards should be treated as C3, or essentially uncontrollable.

This assessment leads to high functional safety requirements for automated driving hardware and software; ASIL-D for the main safety path is typical. Sim-

plified, this means that no component on the system's main safety-relevant execution path can have a higher failure expectancy of 10^{-8} per hour, or can fail statistically more often than once every 11,704 years. This means that a person who drives 2.5 h each day for 50 years has a probability of ∼0.1 % of ever experiencing such a hazard, which is certainly a noble but also very high goal.

ISO26262 gives some guidance on how to achieve this goal for electric, electronic, and software quality levels. But highly automated vehicles base their decisions on sensors and algorithms. Experience shows that most often hazards are not caused by electric or electronic problems, and not even by software bugs (although those are still much more frequent than hardware ones), but some algorithm coming to the wrong conclusion about what it sees in the sensor data. If the automation makes a mistake that is not due to electric/electronic or software error, and everything works as it was tested, but the algorithms make fatal mistakes, that is not acceptable for level 3+ systems.

Recent developments have therefore been focusing not only on functional safety, but on safety of the intended functionality (SOTIF). This can be seen as an extension of ISO26262 which takes algorithmic and perception hazards into account by taking an even more birds-eye view of the system. Elektrobit (EB) believes that this is the correct approach for a global assessment of an automated system, but it does not discuss how to actually reach the required safety. This allows maximum freedom in implementation, but it fails to address the state of the art in actual system implementation. EB believes that it is a good idea to begin discussing not only how to set the goals for automated driving safety, but also how to achieve them in practice.

ALGORITHMIC REDUNDANCY

There are many ways to make a system safe, at least in theory. One popular approach on a macro-system level is to develop a "main function" and a "supervisor function", each of which is fed with mutually redundant environment models, and which assert each other the correctness of their current situation assessment, **FIGURE 1**. This method allows for distribution of the functional safety requirements to two ECUs (or two CPUs on one ECU) and two software development teams.

MODELING THE WORLD

There is great potential for this kind of redundancy on a much deeper and more detailed level. Let us consider, as an example, the time to collision (TTC) that an automated driving system must calculate. The most important input to any safety-relevant vehicle control function will always be "you are headed for a collision in x seconds". In many current setups, TTC is calculated using an environment model known as object fusion.

The idea behind object fusion is that most potential collision partners in the car's environment are either static or dynamic, and their motion (if any) can be modelled with a certain mathematical function. Another vehicle, for example, typically has a steered axle and a powered axle, and moves according to the angle of the former and the speed of the latter. It can therefore be expected, for example, not to suddenly jump to the side or instantaneously stop, but is expected to move according to a specific motion model. This property can be exploited to predict the object's motion, and to filter out measurement error. The typical cycle of an object fusion component is therefore, **FIGURE 2**:

- predict an object's position
- associate the predicted object with a new measurement from the sensor
- update the object's motion model to correspond to the new sensor information.

This type of modelling is typically done with a Kalman Filter [1] or some variant thereof, a 1960 method that has been used in robotics research since the 1980s for object motion modeling. It can be considered tried, tested and state-of-the art, and is used in almost all automotive environment sensors. If the assumptions about the chosen motion model are correct and all error sources in the system are modeled correctly, the result is provable to be the best way possible to describe the object. If this algorithm is implemented correctly, therefore, the system is as safe as can be. Right?

Not necessarily. Object modeling rests on a number of assumptions: The motion model must be the correct one. The engineers must, for example, not confuse the car they are modelling with a different type of object; if it were a pedestrian, it might actually suddenly jump to the side. Since experts have not modelled this, it would surprise their algorithm greatly, it would consider it an error and not tell our function about it. Also, the association must be correct. If they erro-

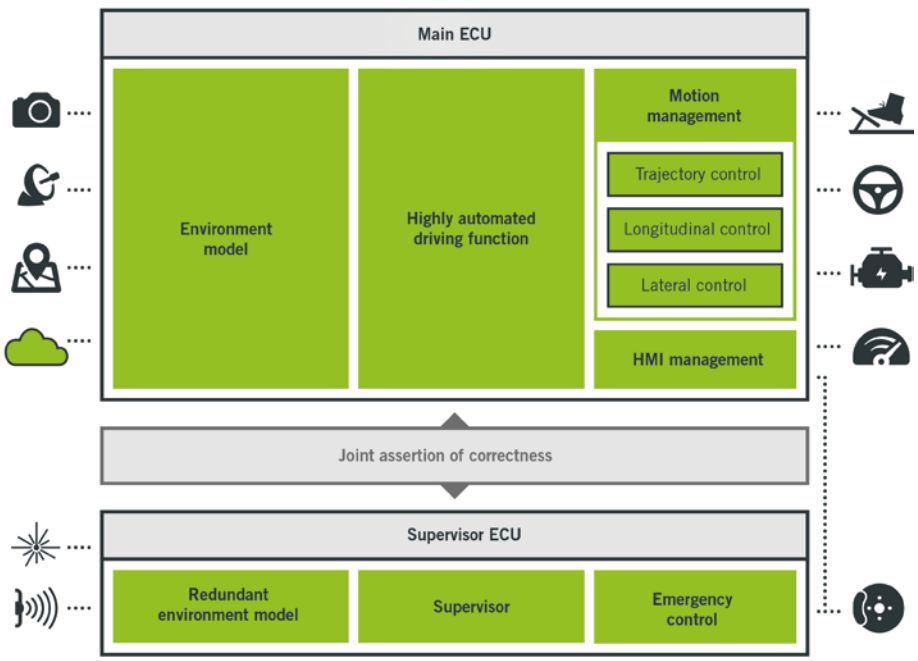

FIGURE 1 The division into main function and supervisor function distributes the functional safety requirements to two control units (© Elektrobit)

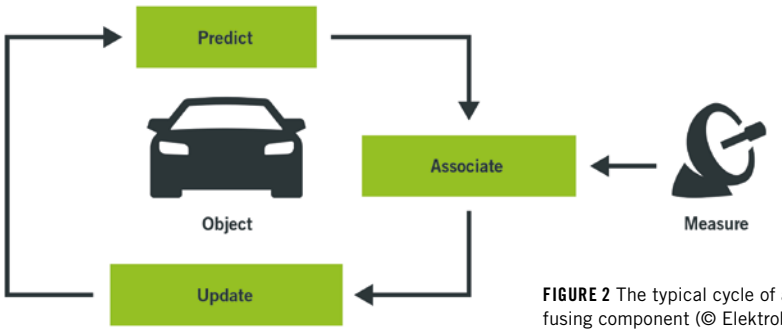

FIGURE 2 The typical cycle of an object fusing component (© Elektrobit)

neously associate a measurement of a lamp post standing beside the road with the modelled car, it would introduce a yawing motion to the car that it is not actually exhibiting. All errors must be modelled correctly. And so on.

That is not to say that object modelling is the wrong tool in this situation. But because it is based on assumptions that can sometimes be violated, it might be occasionally wrong regardless. None of these assumptions are covered by any functional safety standard. The logical conclusion to this problem is to back it up with a redundant algorithm, also state of the art, that delivers the same information in a different fashion.

One other way of modeling the car's environment is to think of it as a grid of discrete cells of a certain size, each of which can be occupied, free, or unknown because it has not been observed yet by the vehicle's sensors, **FIGURE 3**. When a sensor observes an obstacle at a given distance and angle to the vehicle, the grid cell at the corresponding location is marked as occupied, and for certain sensor types the grid cells between the sensor and the observed obstacle are marked as free.

This way of environment modeling was introduced in 1985 [2]. It also is tried and tested in many automotive environment sensors, if perhaps not quite as widely used as the Kalman Filter. It can also provide information on time-to-collision (TTC) since it also models obstacles in the vehicle's path. It also rests on assumptions, like an area being free if sensor rays can pass through them (which depending on the sensor and object type may not be the case), and the dynamicity of the environment, i.e. how many measurements must be made to a single cell before it is recognised as free or occupied. But these assumptions are fundamentally different than those used in object modeling, **FIGURE 4**.

ALGORITHMIC REDUNDANCY CREATES SAFER SYSTEMS

EB has just examined, as one example, two widely used mechanisms to model the environment. Both are well-tested in the field, and can be developed to arbitrarily high functional safety levels (at arbitrarily high development cost). Neither will be correct in all cases. If functional safety and SOTIF demand a failure rate of less than once per 108 operating hours, it is safe to assume that in this operating time either of these algorithms will produce failures, for the prevention of which neither functional safety or SOTIF gives any guidance. The whole system has many such examples. That is a big problem for an automated system which can harm humans if it makes errors in judgment.

One way to prevent these judgmental errors is to use multiple methods for modeling the environment at the same time, and only allowing an action if all of these methods agree on its safety. In the example above, our vehicle might only continue driving if both the "object" model and the "freespace" model of the environment both agree that there is nothing in the vehicle's path. Since the assumptions behind both methods are fundamentally different, there is hope that they are not both wrong at the same time.

THE NEED FOR PEER REVIEW

But autonomous vehicles should not be hoped but known to be safe. SOTIF does not specify decomposition rules for algorithmic assumptions, but it does specify that they cannot be wrong more than once every 108 h for an ASIL-D system. This is true for other types of system as well (e.g. brake mechanisms), but in our case the problem is much less understood and cannot be tested in the lab. Therefore the necessary certainty can currently be only reached by field testing, which would require a number of statistical tests, one of which takes 108 h. It is prohibitively expensive to draw even one such sample by test driving a real car, let alone the number of samples required for

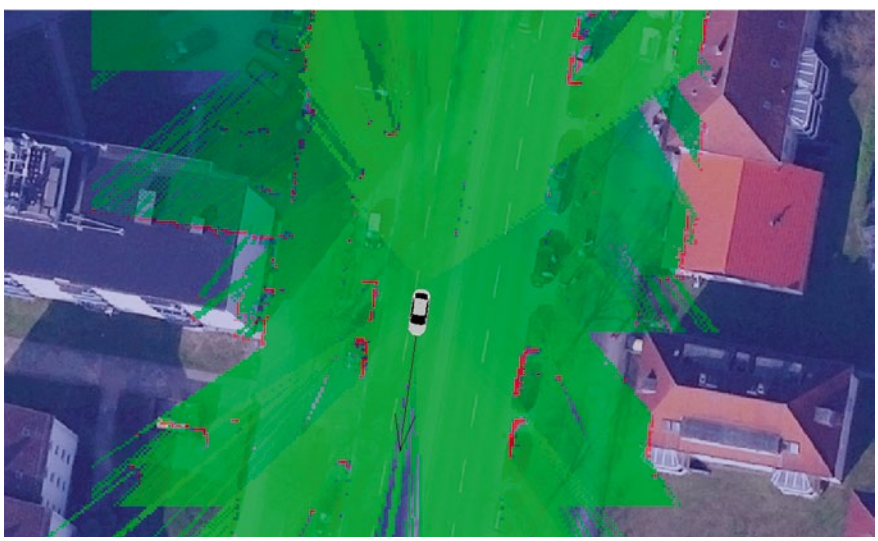

FIGURE 3 A network of free, occupied or unknown cells forms the environment of the vehicle (© Elektrobit)

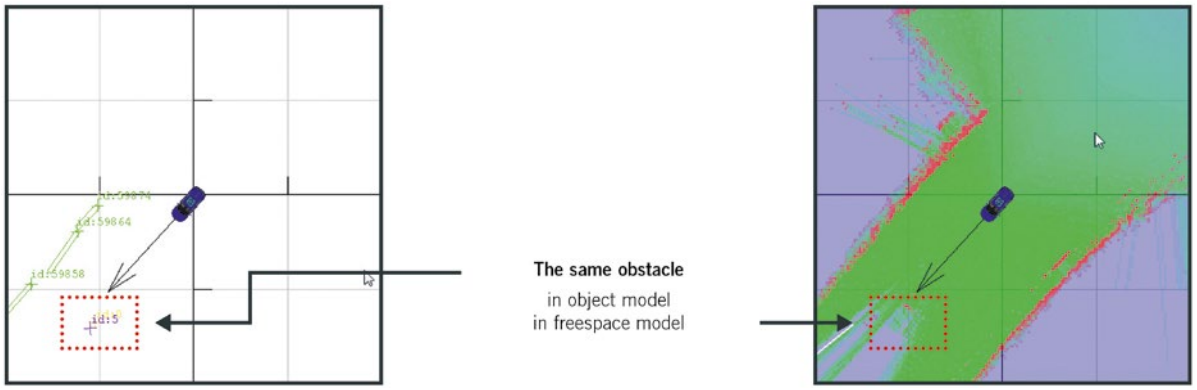

FIGURE 4 Differences in the object recognition in the object model and the freespace model (© Elektrobit)

statistic certainty. Test driving is therefore, scientifically speaking, poking in the dark. There has certainly been very good experience with this in regular assisted driving, but given the enormity of the ISO requirement on system safety it can currently not be guaranteed that testing is enough. This means that there currently is no technical way of approving the safety of any algorithmic solution for automated vehicles.

There are ways around this. In science, it is tackled by peer review: A given method, which is believed to solve a problem, is published and described in detail. It is examined by others, and flaws are exposed by theoretical examination and practical testing. Refinements are found, published again, and re-tested. This is typically a cycle for any given solution, towards the end of which there are often many slightly different solutions, which are then compared. In the end, a gold standard emerges which stands for some time, until intelligence or advances in other fields bring new ideas to the table, and the cycle begins again. Both algorithms described above have gone through this cycle and are considered the gold standard solution to their specific problem. Peer review does not guarantee a 100 % safe solution, but it certifies that at a given point in time, no-one has thought of a better solution.

In the race to be first to market, development in highly-automated driving seems to go the other way. Certainly gold-standard solutions for specific problems are used, but their combination into complete systems is treated as company intellectual property and hidden from view. The public marketing of recent advances in neural network research-seems to suggest that even using gold-standard algorithms is not à la mode any more, and instead artificial intelligence will solve the problem by some undisclosed magic. This is a very short-term solution. Certainly at some point in time a human will be harmed in an accident with highly automated systems, whether it is the system's fault or not. In that case, a court examination will try to find out whether the system has been developed to the state of the art. If at that time no state of the art has been established, it can be assumed that uncomfortable discussions will be had by many. Depending on the court's judgment, and possibly ensuing lawmaking process, this can not only harm the affected companies but the entire automated driving industry, estimated to be a 42 billion US dollars market by 2025 [3].

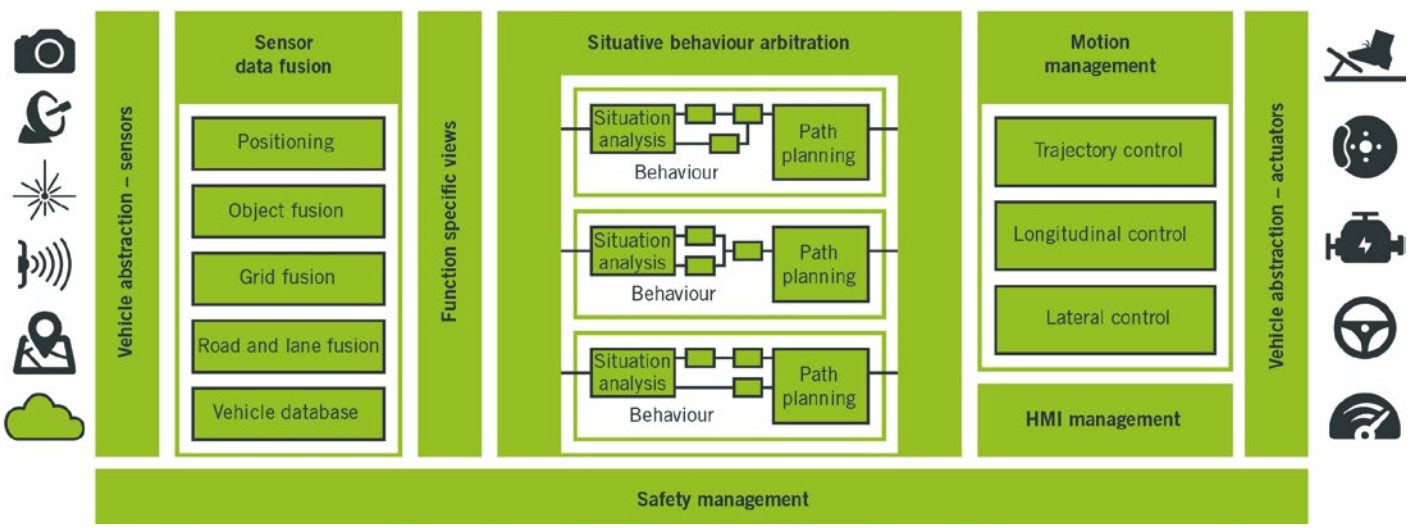

FIGURE 5 the software solution EB robinos is based on „open robinos", with open interfaces for highly-automated driven systems (© Elektrobit)

THE NEED FOR STANDARDS

EB therefore proposes to begin a discussion in the industry on best practices for the construction of highly automated driving systems, on a deeply technical level. It is not necessary that manufacturers disclose their own solutions in depth (although it would ultimately be desirable), only that there exist a published solution that can be peer-reviewed and improved upon over time. In our opinion, this does not endanger any manufacturer's intellectual property in any way.

In June 2016, Elektrobit has published a description of their own solution under the name of open robinos [4], which has been implemented to run on standard PCs and some automotive ECUs, **FIGURE 5**. The specification (i.e. the system description to be peer-reviewed) is available under a Creative Commons license, and the implementation is freely available. EB is currently building a consortium around this specification which will start the reviewing process, and publish improvements to the specification. The goal is to create a reference implementation other systems can be compared and tested against, and eventually build an open standard for constructing self-driving systems. A standard has other implications than only peer-reviewing safety-critical algorithms. It opens the door for software, hardware, and vehicle manufacturers to use parts of the standard, and replace others with their own solutions. It opens a way to develop own IP or brand-shaping features without the need for developing the entire system. It creates a market for system modules, so competition can happen on a per-module and not on a per-system basis. This creates better technological solutions, and spreads development and testing cost across an industry instead of only a car model. A standard can be discussed with lawmakers, used to create testing procedures for agencies like NHTSA or TÜV, and employed to guarantee to the customer that a product adheres to it and is therefore considered safe to use.

CONCLUSION

In this paper, EB has not published a new technical solution. It is a very old solution to the problem of finding the right technological advances, which is currently not employed in the race towards self-driving cars outside of university research. EB is convinced that currently the problem of highly automated driving is not understood well enough for a single company or conglomerate to solve on their own, and even if it was, that this would not solve the problem of creating safe self-driving cars as a whole. At least the principles of automated driving need to be reviewed, tested, approved, and ultimately standardised by the industry.

The presented suggestion does not detract from any company's business, or impede the development speed for self-driving cars. On the contrary, EB believes it will allow better use of monetary and personnel ressources, speed up the development process and make safer systems.

REFERENCES

[1] Welsh, G.; Bishop, G.: A Gentle Introduction to the Kalman Filter. In: Proc. SIGGRAPH, August 2001

[2] Moravec, H.; Elfes, A.E.: High Resolution Maps from Wide Angle Sonar. In: Proc. ICRA, March 1985

[3] Green, J.: Driverless-Car Global Market Seen Reaching $42 Billion by 2025. Online: https://www.bloombergcom/news/articles/2015-01-08 driverless-car-global-market-seen-reaching-42-billion-by-2025

[4] Elektrobit GmbH: Open robinos specification. Online: https://www.elektrobit.com/products/eb-robinos/eb-robinos-specification/