Compliance, Functional Safety and Fault Detection by Formal Methods

Christof Fetzer, Christoph Weidenbach, Patrick Wischnewski

Compliance, Functional Safety & Fault Detection by Formal Methods

Christof Fetzer
Technical University Dresden, Germany
christof.fetzer@tu-dresden.de

Christoph Weidenbach,
Max Planck Institute for Informatics, Saarbrücken, Germany
weidenbach@mpi-inf.mpg.de

Patrick Wischnewski
Logic4Business GmbH, Germany
patrick.wischnewski@logic4business.com

October 10, 2016

Abstract

With the increasing complexity of today's cars functional safety and compliance guarantees are more and more difficult to obtain. During the life time of a vehicle the detection of malfunctioning non-mechanical components requires meanwhile more attention than the maintenance of its mechanical counterparts. A full fledged formal verification of the overall car is not realistic and even hard to obtain for single non-trivial components such as assistant systems. Furthermore, it does not support fault detection at run time. We suggest an approach towards formal safety, compliance and fault detection at run time via an auditor. The auditor is automatically fed out of the engineering and production process by a suitable abstract specification and respective model of the car and can detect then detect violations and faulty components.

1 Introduction

The big advantage of cars as a system compared to software is robustness. A single spark ignition failure during an engine run is hardly ever recognized

by the driver nor does it significantly influence the behavior of an engine. In contrast to the robustness of an engine, a single faulty line of code in a big piece of software typically causes undesired behavior of the overall system. On the other hand and in contrast to software, all parts of a car can unexpectedly break at life time. Therefore, instead of a rigid formal verification of a car, the robustness offers another possibility: the verification of the car's behavior with respect to an abstract specification and model at run time. The approach is not unrealistic. Car manufactures are typically able to predict the behavior of their cars with respect to such an abstract model on the basis of a few hundred specification parameters. For example, acceleration from 0 to 100 can be predicted up to an accuracy of at least 5% already at the design time of a new car. Similarly, emission values of fuel consumption can be predicted as well.

This motivates our approach to guarantee functional safety, compliance and fault detection. An auditor is added to the car. It reads the run time parameters while driving, knows the abstract model and specification of the specific car, and may obtain further information, e.g., GPS information, by communicating to the outside world. The auditor compares the run time parameters with the abstract model. In case of a violation beyond the accuracy of the abstract model, it can act accordingly. Online monitoring has been already investigated previously, for example, in the context of a gear box [5]. Also, some auditing functionality is today already build into modern cars, e.g., for engine safety, our approach is generic in the sense that it can be applied to functional safety, compliance and fault detection, in general. Furthermore, because it is built on top of a formal model through engineering and manufacturing, it can even consider parameters at the level of a specific car. Any property that can be supported by a sufficiently accurate abstract model can be subject to the auditor. Furthermore, while the currently deployed auditors are hand-coded, we indicate that our auditor approach can be actually automatically generated out of an engineering and manufacturing process supported by an overall formal model. Because it is automatically generated, it can even be more accurate and potentially cheaper than today's approaches. For example, it might be able to distinguish hardware from software failures at run time [4] and may consider car specific properties such as electricity consumption.

The paper is now organized as follows: Section 2 explains how the formal abstract model and needed input parameters can be obtained out of an engineering and manufacturing process supported by formal methods. In Section 3, we explain our auditor approach, where a generic auditor is fed by the parameters and models from Section 2. The paper concludes with a

discussion of the obtained results and its potential impact in practice.

2 Development

All car manufacturers already have a logical description of the cars they can build, they can produce, and they want to sell. This description is typically based on the parts of the car, including software at the same level as the mechanical parts, and also contains key specification properties. From the engineering process abstract models representing specific aspects of the overall car are available as well. For our approach, we accumulate all this information into one formal model.

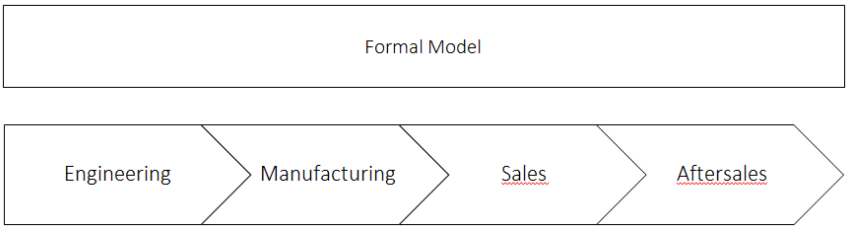

Figure 1: Formal model representing all product variants over the product-life-cycle

We propose a rigid development, manufacturing and sales process that is build on top of a formal, consistent overall formal model representing all eventual products. Aftersales can make use of this model at a car specific level. The model is based on the *parts* of a car. Attached to parts are *attributes*. *Rules* in form of formulas of an appropriate logic state how the different parts can be combined to *components*, and eventually to *products*, i.e., cars. Furthermore, *constraints* on attributes, assigned to components and products further enable the fulfillment of properties beyond the bill of material.

For example, a *component* may be a particular engine, that contains as its *parts* a specific block, cylinder head etc. It carries as its *attributes* the overall volume and number of cylinders. Together with further parts, including a particular engine software, transmission, etc., the engine becomes an application for a particular car. Its emission values are measured in the respective test-stand cycle and become *attributes* of the application and eventually parameters of the final abstract model.

Typical constraints with respect to engineering, manufacturing and legal requirements could be:

- the size of the gasoline tank of the car stays in a sound relation to its fuel consumption and desired reachability,
- the engine software properly matches the number of cylinders of the engine and considers the attached turbo charger, and
- the fuel consumption, and hence respective emission values, is limited at particular state of the engine, determined, e.g., by its current load,

respectively. If all this information is in fact the basis for the overall building process of the car, the parameters of the auditor for a particular car variant can be automatically set during manufacturing.

A formal model representing all product variants with its attributes comes with the big advantage that formal verification becomes available early in the process and can be used in order to prove that all specified rules are fulfilled. Consequently, the proof serves as a certificate for the correctness of the formal model in terms of safety, manufacturing and compliance requirements and the parameters for the auditor are automatically extracted from the certificate. With these parameters, the auditor observes and, hence, ensures that the correctness of the formal model is properly implemented and kept during life time in the real car. Consequently, the correctness of the overall system (car) is ensured over the complete life-cycle.

Furthermore, building, maintaining and automatically deriving properties of such an overall model of all car variants and its attributes provides the opportunity to perform optimizations on the formal model with respect to a wide range of parameters. This is not completely beyond scope of today's performance of automated reasoning tools. In [6], it is shown that the emission class of a car can be automatically obtained out of a model of the above form. For example, the weight of the car, one input parameter of the emission class formula, is accurately computed in this abstract model out of weight of the different parts of the car. It is even shown that optimization problems, e.g., answering questions of the form "what is the cheapest car with emission class x that contains the components y, z, ..." can be solved by appropriate usage of automated reasoning technology.

Although a rigid management of the product built process is a well known topic in the engineering and manufacturing science [3], an approach based on an overall formal model has not yet been developed. There is work on certain aspects of a formal model build on top of bill of material for the

product [7] but it does not consider complex constraints and attributes as suggested here. These are needed to eventually and automatically provide an abstract model and specification to an auditor. A first step into this direction is described in [6].

3 Auditor

The objective of the online auditor is to monitor at runtime if all requirements (e.g., clean air regulations) are satisfied. To be able to do so, the car and in particular, its components like the engine controller, have to provide the auditor with all information that are needed to perform this real-time compliance check (see Figure 2). The auditor has a formal model of the car together with its specific parameters that permits it to perform this compliance check, e.g., to ensure that emissions regulations and safety regulations are satisfied. This formal model is a result of configuration information generated in the development process (see Section 2).

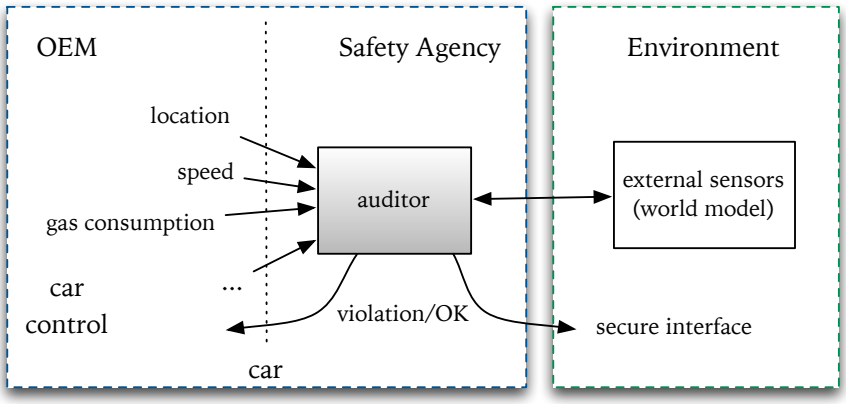

Figure 2: The online auditor runs inside the car. The car provides the auditor with real-time information that permit it to check if requirements are satisfied. The auditor has access to external sensor data to crosscheck and to extend the real-time information.

The idea is that the auditor computes the compliance information and reports not only to the OEM but also during the next motor vehicle inspection if there were any violations. This report is certified to show that this report was generated by a valid auditor. An inspection station is permitted

to download and analyze this report, in particular also for the detection and repair of faulty components.

Note that the auditing of the emission data could in principle be performed outside of the car. For example, a car could upload all necessary information in real-time to some cloud service. This might, however, expose privacy relevant data to the cloud and might, moreover, require constant Internet connectivity. Similarly, if we would store all sensor data, one could process the data, say, at an inspection station. However, this would imply that all data must be stored for long durations ($>$ 2years) - which would be too much data to store.

An online auditor does not need to store massive amounts of data nor does it need to send data across the Internet. It has to store only minimal amount of data: any violation of regulations is stored persistently until the next motor vehicle inspection. The auditor could even be configured to report violations immediately, e.g., to ensure that safety violations are immediately addressable. The auditor will not reveal any private data during normal operations. The auditor is the result of a formal car model, therefore, it can even be verified whether private data of the driver or intellectual property of the OEM is leaked to the outside. However, if a violation of some regulations has occurred, this information needs to be verifiable: the location of this violation might need to be reported as well as all sensor data relevant for proving that a violation has happened.

If a violation can be shown based on the information provided by the car, the auditor needs to show that the sensor data was indeed been provided by the car. Hence, we require that the car certifies all sensor data and the auditor only accepts certified sensor data. In this way, the auditor could formally prove to an external auditor, that a violation has happened. This proof could also be used to provide evidence to car manufacturers to fix the detected issues.

If a car would consistently lie to the auditor, the auditor might never detect a violation since a good liar will, of course, appear to satisfy all requirements. Since the auditor does not have any sensor itself, it cannot use its own sensing to detect wrong sensor information. To address this issue, we permit the auditor to read external sensor information. Note that the use of external sensor information requires constant Internet connectivity. The external sensors might be received from other cars but also from the smart street sensors or some sensors from, say, an inspection station. The external sensor information is combined to a "world model" by an external service and certified. The world model contains real/time information and information regarding the accuracy of this information. Contradiction be-

tween the internal sensor data and the external sensors data will be reported as a violation.

Note the the auditor has similarities to the *monitoring functions* used in ISO26262 [1]. Monitoring functions can be used to detect errors during operations. Any error that is detected will trigger some recovery action. An auditor is a more general mechanism since it cannot only used to detect errors but also monitor the compliance with various regulations. Moreover, an auditor can be executed in an hostile environment. Besides compliance monitoring and error detection, auditors could also be used to support the *field monitoring process* [2]. Field monitoring is used to collect data about any potential safety violation during operations. This cannot only be used to detect issues in safety critical software but also to support *proven in use arguments*.

4 Conclusions

Technical systems are highly complex and the verification that they are functionally safe and in compliance with all applicable regulations is very difficult to ensure using design time verification. In addition, design time verification does not support run time fault detection. Our proposal is to perform a continuous verification of the compliance of a technical system with its specification at run time. This should ensure a level playing field for all manufacturers - independent of their interpretation of legal regulations. Our approach balances regulatory compliance, protection of intellectual property by the manufacturer and the privacy of the users. It requires a formal-methods supported development process by using artifact created during the development with the auditor-based compliance checking. The auditor protects the IP of the manufacturer by processing in real-time all data and only keep data in case a violation of regulation was detected. Moreover, the auditor protects the privacy of the car user. Only in case of a violation, anonymized data may be forwarded.

We can think of a wide range of potential instances of the auditor. It may be an external component that is only attached to the car by a regulation authority to check compliance. It may be an external component used by the garage for fault cause detection. It may be a permanent part of the car and generalize the mechanisms already in place today.

References

[1] ISO/DIS 26262-1 - Road vehicles Functional safety Part 1 Glossary, July 2009.

[2] ISO/DIS 26262-7 - Road vehicles Functional safety Part 7 Production and operation, 2009.

[3] H. ElMaraghy, G. Schuh, W. ElMaraghy, F. Piller, P. Schönsleben, M. Tseng, and A. Bernard. Product variety management. {*CIRP*} *Annals - Manufacturing Technology*, 62(2):629 – 652, 2013.

[4] Majdi Ghadhab, Matthias Kuntz, Dmitrii Kuvaiskii, and Christof Fetzer. A controller safety concept based on software-implemented fault tolerance for fail-operational automotive applications. In Cyrille Artho and Peter Csaba Ölveczky, editors, *Formal Techniques for Safety-Critical Systems - Fourth International Workshop, FTSCS 2015, Paris, France, November 6-7, 2015. Revised Selected Papers*, volume 596 of *Communications in Computer and Information Science*, pages 189–205. Springer, 2015.

[5] D. Heffernan, C. Macnamee, and P. Fogarty. Runtime verification monitoring for automotive embedded systems using the iso 26262 functional safety standard as a guide for the definition of the monitored properties. *IET Software*, 8(5):193–203, October 2014.

[6] Christopher Junk, Robert Rößger, Georg Rock, Karsten Theis, Christoph Weidenbach, and Patrick Wischnewski. Model-based variant management with v.control. In Richard Curran, Nel Wognum, Milton Borsato, Josip Stjepandic, and Wim J. C. Verhagen, editors, *Transdisciplinary Lifecycle Analysis of Systems - Proceedings of the 22nd ISPE Inc. International Conference on Concurrent Engineering, Delft, The Netherlands, July 20-23, 2015*, volume 2 of *Advances in Transdisciplinary Engineering*, pages 194–203. IOS Press, 2015.

[7] Marcílio Mendonça, Andrzej Wasowski, and Krzysztof Czarnecki. Sat-based analysis of feature models is easy. In Dirk Muthig and John D. McGregor, editors, *Software Product Lines, 13th International Conference, SPLC 2009, San Francisco, California, USA, August 24-28, 2009, Proceedings*, volume 446 of *ACM International Conference Proceeding Series*, pages 231–240. ACM, 2009.