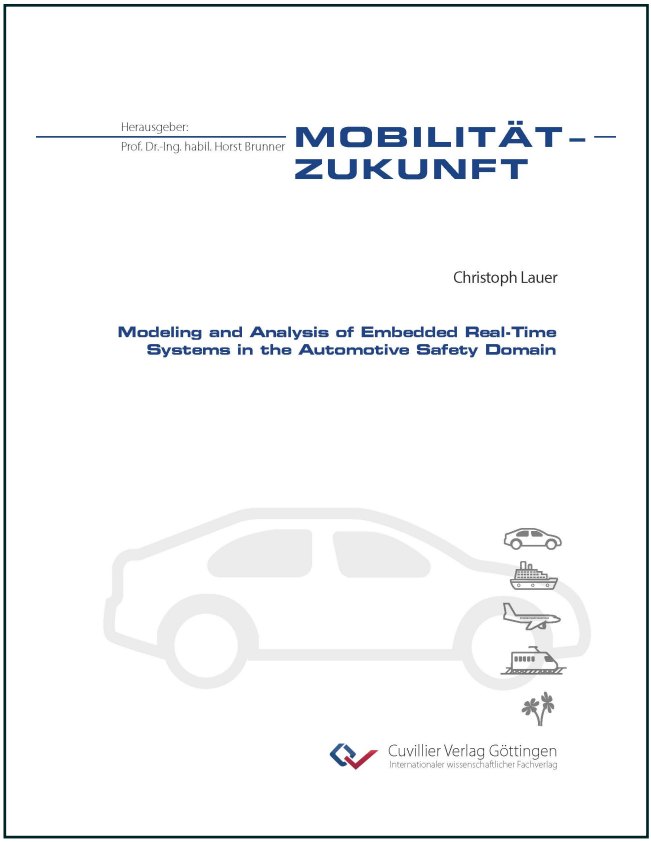

CHAPTER 1

Introduction

Ever since the development of the first integrated circuits in the late 1950s the complexity of such devices doubled every 20 months. A development which has been anticipated by Gordon Moore in 1965 and has been confirmed by the Intel Cooperation to remain valid until 2029[1]. The very large scale of integration in these circuits results in more efficient and powerful devices while the mass production of integrated circuits made them available for industries such as the automotive industry. In the past decades this development led to a wealth of potential applications that use the increased computational, communicational, and sensing capacities to add intelligent and context sensitive behavior to the automobile. Implementing such context sensitive applications has system wide implications and an inherent complexity that imposes major challenges on OEMs as well as suppliers. This chapter is intended to give the reader an overview of the latest developments in the automotive safety electronics domain and rational for the application of model based analysis methods particularly at early design stages.

[1] As mentioned at the 2008 Intel Developer Forum in Shanghai

1.1 Trends in Automotive Safety Engineering

The current state of vehicular safety can be described as transitional between the purely mechanically constructed safety from the past and computer supported autonomous driving that takes the human factor out of the equation of fatal road accidents. In the past, before the increased use of electronics in automobiles, vehicular safety was asserted by a robust chassis construction, safe seat construction and passive restraint systems, such as safety belts. For a long time safety engineers focused on building and improving these passive safety systems and adding more sophisticated passive safety systems. Today researchers and engineers dream of autonomous driving and zero-casualty road traffic. The goal is to equip automobiles with sensing capabilities superior to the human eye, decision making capabilities superior to the human brain, and maneuvering capabilities superior to those of a human driver.

The first electronic based safety systems have been mass deployed in 1971 after the pass of a respective law in the U.S.[2]. As a result reactive airbag systems were built, that inflate within a blink of an eye in case of an accident, and prevent the driver from hitting the steering wheel. In the following decades the Anti-lock Braking System (ABS) and Electronic Stability Program (ESP) were developed and a remarkable reduction of traffic casualties was achieved albeit a massive increase of road traffic in the years after. In recent years the actual number of road fatalities in developed countries has not increased although the number of licensed cars continually increased (see Figure 1.1). To continue this positive trend today, cars are equipped with sophisticated remote sensing devices for environment perception which enable the vehicles to act in a context sensitive manner and pro-actively mitigate hazardous driving situations. The deployment of the sensing devices, control, communication, and actuation systems transform the purely mechanical automobile from the 19th century into a high-tech vehicle more complex than early aerospace vessels. Still, in 2004 the WHO

[2]National Traffic and Motor Vehicle Safety Act (1966)

decided to drive further reductions of the number of people killed in road traffic, particularly in developing countries, by means of technological improvements [PSS+04].

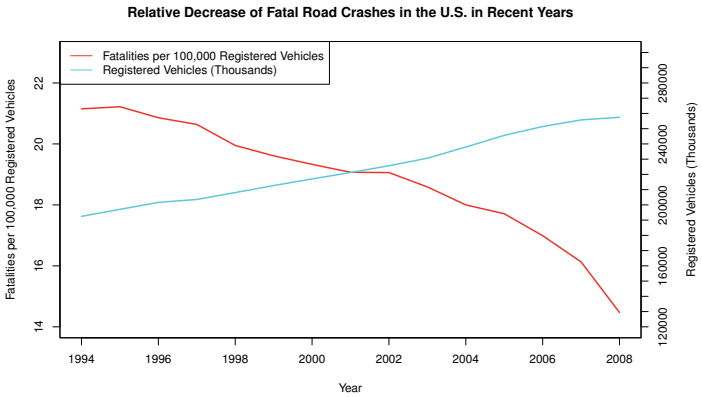

Figure 1.1: In recent years the number of fatal road crashes relative to the number of licensed cars kept declining in the U.S.; Source: Department of Transportation, NHTSA FARS Encyclopedia.

The physical space to deploy new electronic components is limited[3]. Also, new devices increase the weight and electrical energy demand of the vehicle. Therefore, one strategy to deploy new applications is to integrate them into existing components with more powerful controllers. These domain specific controllers are sophisticated computing machines that implement several application instances which belong to the respective application domain, e.g. body, chassis, infotainment, and safety. Clearly, the complexity of the distributed system remains manageable while the complexity of the nodes in the distributed system increases [POT+05]. In addition to technological challenges that arise from the increased functional integration, sociological and legal issues come into play when multiple suppliers

[3]Figure 1.2 illustrates the complexity of the wiring of a middle class automobile

and multiple departments of the Original Equipment Manufacturer (OEM) itself deliver components of the integrated device. This is a trend that can be observed already and will increase as the value creation of an electronic based application is very high [Ins04, DH08].

Figure 1.2: The wiring and deployment of new electronic components is difficult because existing components maintain large parts of the available physical space.

1.2 Technological Trends in Safety Electronics

Aside from the sensing components dedicated to environment perception, e.g. Radio Detection and Ranging (Radar), laser scanner, video camera, and Photonic Mixer Device (PMD), the computational interpretation of

the acquired data plays a key role in pro-active safety systems. Object detection, trajectory estimation, and object classification are some of the most prominent tasks to be performed by the domain specific controllers. Moreover, sophisticated collision detection, crash mitigation, and action concepts have to be evaluated in a timely and safe manner. Each millisecond lost on the way from the sensor data acquisition to the respective actuator, like brakes, alarm, and traditional safety systems, reduces the effectiveness of the respective pro-active action. Since most of the mentioned tasks are implemented in software this represents a life critical real-time system with high levels of real-time requirements and Safety Integrity Level (SIL).

Not only the developments in the safety domain but also in the chassis domain, where the *X-by-Wire* approach aims at implementing the basic operation tasks of steering, braking, and throttling by electronic components, led to the adoption of an architectural approach called the Time Triggered Architecture [Kop97, Kop98]. In a time triggered embedded system pre-defined schedules coordinate the instants in time when a specific action is allowed to be performed. Such architectures offer timing determinism at the design phase which is particularly interesting for many safety critical applications. The scheduling of communication messages, however, impose challenges on the task scheduling on each controller, especially if the controller is implemented in a synchronous time triggered manner as well [Ric07, Ric08]. In contrast to traditional event triggered systems, timing requirements for high priority tasks are more difficult to understand, thus leading to engineering challenges the automotive industry has not faced before.

Safety analyses on the other hand are common practice in this field. With the release of the ISO 26262, a new development and validation standard for automotive safety applications, process guidelines will become more specific. For software systems as described above the standard demands the development of the complete system with the maximum safety requirements based on the most critical component or the validation of a so called *freedom-of-interference* between all software components, i.e. a proof that a failure of one component may not cause the failure of another component. Another technique proposed in the standard is the hierarchical decomposi-

tion of components where redundant decision making may be used to implement a higher level of system safety. The discipline of functional safety, i.e. understanding and mitigating safety risks plays a major role in product design in general and has a large impact in safety engineering in particular.

Most Electronic Control Unit (ECU) design goals do not contradict or exclude each other per se. However, the automobile is a complex distributed system which has to be considered from many different view points. Often certain design decisions have implications on areas of product design that appear completely independent from one another. For example, increasing the computational capacity of an embedded controller for software timing reasons may require a larger energy reserve which is required for safety reasons. A larger energy reserve comes with larger power supply electronics, thereby, increasing the ECU dimensions. ECU dimensions play a major role for example for physical design decisions, hence, software timing issues may eventually have an impact on the physical design of the car or, which is more likely, the other way around. As a result of the numerous side-effects of different system design domains, design decisions change rapidly especially at early design stages and the developers face many uncertainties.

1.3 Model Based Analysis

The design of a complex integrated ECU is a difficult task that is under the influence of various side effects from other design decisions in different modules of the automobile. Especially at early design stages this results in rapidly changing design concepts that may differ significantly. Design decisions may include the actual number of ECUs the system consists of, the topology of the intra-car network, the communication technology between the integrated ECU and the rest of the automobile, operating system paradigms, controller architecture and technology, and the number and nature of algorithms which will be implemented on the domain specific controller(s).

At these early stages, when changes in the system design are easier and less expensive to accomplish than later in the development process, the developers require an in-depth understanding of the important perfor-

mance measures. The definition of the performance measures depends on the focus of the developers. An OEM has different departments for chassis, body, safety, infotainment, and power-train development, but even within the departments itself a strong sense of hierarchical organizational structure is implemented. This results in dedicated departments that are responsible, for example, for safety ECU design, safety application development, communication system development, software architecture development, and sensor system design. For the ECU design non-functional properties, like package dimensions, controller capacity, software timing, and costs are most important. Functional properties play an important role for application developers but may also be important for the ECU design if a particular implementation requires dedicated components which have an impact on the overall architecture. As stated before, several side effects from seemingly uncorrelated development domains exist.

Although it is important to understand all implications and bottlenecks as early as possible in the development process, the time available for analysis and evaluation of architecture concepts is scarce; A dilemma. This dilemma can be solved by appropriate analysis methods which help to abstract from unknown system properties and allow for automated system evaluation. The discipline of embedded systems engineering has established a wealth of methods for modeling systems on different levels of abstraction and evaluating both functional and non-functional system attributes. Besides the selection of adequate performance measures, modeling and analysis approaches care has to be taken in terms of how to generalize the analysis results as every modeling approach incorporates an inherent abstraction from the real system.

CHAPTER 2

Modeling Scope and Research Question

Modeling and analysis is an often used design methodology when complex systems are involved. Particularly at early design stages when certain components may not yet be fully specified, abstraction is a feasible concept to investigate the system without having to implement it. Depending on the scope of the analysis the model of the system has a specific scope, too. One classification used in this thesis allocates modeling and analysis methods to functional and non-functional properties-:

- Functional modeling: Sommerville [Som07] defines functional properties as properties that describe "what the system should do". Functional properties of an ECU include the output values of an application hosted by an ECU given its current state and input values. An example for a functional property in the automotive safety domain is the correct detection of an imminent crash based on remote sensing devices.

- Non-functional modeling: A non-functional property describes "how a system is", i.e. a Quality-of-Service (QoS) attribute of the system with respect to a service domain different from the system's functional purpose. These domains may include reliability, costs, physical

dimensions, timeliness, maintainability and usability. Clearly, systems with the same functional properties can differ significantly in terms of their non-functional properties.

According to Sommerville's definition of functional and non-functional properties a measure that captures how well a system performs, for example in terms of measurement accuracy, false positive, and false negative measurements, is a non-functional property. However, to acquire this kind of performance measures may require a functional model, that is, a model describing what the system is expected to do.

The concept of system modeling incorporates an inherent inaccuracy since the model is, by definition, an abstraction from the original system. Depending on the scope of the analysis the model abstracts from unrelated system attributes and captures all relevant attributes. Non-functional models abstract from the functional details of the system whereas in functional models the non-functional attributes of the system is abstracted from. An important part of any modeling approach is the input modeling which describes the characteristic of the system variables. The input variables of a system play a major role in any analysis method, hence, the quality of the input models directly affects the quality of the system model as such. Both the quality of the input variables and the degree of abstraction of the model have to be considered when generalizing the analysis outcomes. If aspects of the system, that may influence the observed attribute, have been abstracted from, then the results of the analysis have to be generalized taking into account potential side effects. The same is true for the modeling inaccuracy of the input models.

The actual model representation and analysis method depends on the system as well as the performance measure one aims to evaluate. The outcomes of an analysis can deliver qualitative and quantitative performance measures. Quantitative analysis derives numerical performance results that capture an absolute QoS measure that can be correlated with the results from any other analysis method that investigates the same measure. Qualitative analysis results establish a relation between two or more alternative system models and provide a relative performance measure of the analyzed