

# Modeling Data Requirements for a Secure Data Management in Automotive Systems

Sandro Schulze, Tobias Hoppe, Stefan Kiltz, and Jana Dittmann

School of Computer Science, University of Magdeburg, Germany,  
{sanschul, t.hoppe, kiltz, jana.dittmann}@iti.cs.uni-magdeburg.de

**Abstract.** IT security and data management are two emerging aspects in automotive systems. In this paper we propose a model which supports the integration of these aspects, already starting in the early stages of development. This systematisation of different kinds of data processed within automobiles, including their specific requirements (with respect to safety, security, comfort etc.), allows for a wide range of applications like implementation guidelines and verification. Such holistic approaches are essential for future automotive data management, especially to cope with the increasing complexity and IT security requirements in data management.

**Key words:** Automotive, IT security, data management, requirements engineering

## 1 Introduction

Automotive systems are an evolving area with a rapidly increasing amount of software [1], caused by more and more functionality to be implemented. At the same time the amount of data to be managed by the system is increasing [2]. Furthermore, different kinds of attackers have a variety of motivations in attacking automotive IT [3] in different ways [4]. Thus, more attention will have to be paid to both, effective concepts for IT security and efficient data management during the development of Electronic Control Units (ECUs) [5].

Combining these emerging aspects, IT security and data management, could be useful in several ways, including reliability or confidentiality of data. Moreover, it is indispensable to take both into account if we consider an automotive system as a network providing a number of features to interact and therefore, to interfere with [3].

In this paper, we present an approach, which integrates these new automotive aspects of IT security and data management from the beginning into the development process. Therefore we propose a model, which enables modeling data requirements integrated with aspects of security, safety, data management and comfort. The intention is not to develop yet another model-driven code generator for the automotive domain. This model rather addresses the preliminary considerations of the design or implementation phase, e.g., requirements

analysis. As a result, design recommendations for secure data management may be derived from our model. Subsequently, these recommendations can be used for implementing the respective software, using already existing and upcoming standards and architectures, e.g., AUTOSAR ([www.autosar.org](http://www.autosar.org)).

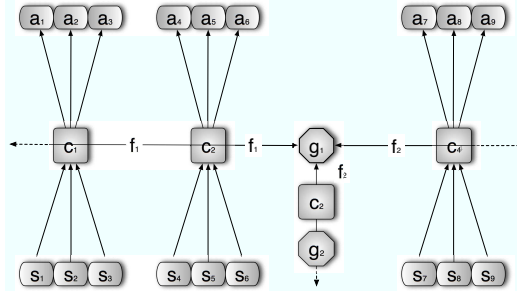
## 2 Integration of security and data management

Our model, presented in the following, is placed within the requirements analysis phase of the software development process. Thus, specifications for the several data (and ECU's), given by an domain expert (e.g. the car constructor), can be taken directly to model the respective requirements.

### 2.1 Fundamentals

Before introducing our model for data requirements, we present a formalisation of the automotive system, representing the environment of the considered data.

As a networked IT-system, we formalise a modern car into the classes sensors ( $s_i$ ), electronic control units ( $c_i$ ) and actuators ( $a_i$ ). Information exchange between cars (car-2-car) or between the car and the infrastructure (car-2-infrastructure) is deliberately excluded within the confines of this paper. The underlying network of our formalisation is depicted in Figure 1.



**Fig. 1.** Formalisation of the automotive IT-network

Each of the corresponding ECUs ( $c_i, i \in \mathbb{N}$ ) gets (analogue or discrete) environmental data from the sensors ( $s_i, i \in \mathbb{N}$ ). With this information the ECU computes values to control the system using actuators ( $a_i, i \in \mathbb{N}$ ). The ECUs exchange information amongst each other by sending data telegrams via field bus systems ( $f_i, i \in \mathbb{N}$ ). Examples for common automotive field bus architectures are CAN, FlexRay, MoST and LIN.

A special variant of an ECU is the gateway ( $g_i, i \in \mathbb{N}, G \subset C$ ), which connects different subbus systems (e.g. powertrain, body electronics, driver information). This includes the decision whether forwarding messages into a different subbus or not and translating data telegrams between different bus architectures.

With the help of the control unit set  $C$ , we define a topology  $T = \{C, Graph\}$ , representing the automotive system as a whole. The graph in this context describes the structure of the system (cf. Figure 1), connecting all control units in a specified way.

Based on this topology, we define a formalisation for automotive data, which has to be extended in future. One important aspect is to integrate a formal representation of the desired data flow. Here, existing data flow models, e.g., Clark-Wilson [6], could be employed. For such purposes, the integration of existing formal models (e.g., from IT security) is not yet addressed by this initial draft, which is explained shortly in the following.

$$D = \{d_1, \dots, d_m\}, d_j = \{Identifier, C_j, Req_j, P_j\}, d_j \in D, \\ C_j = \{c_i \subset C | x_k \in X\}, X = \{R, RW, W\}$$

Within our formalisation,  $D$  encompasses all data within the automotive system. Furthermore,  $C_j$  contains all ECUs accessing a single data  $d_j$  and the way they are allowed to do this, i.e. which rights an ECU has on the considered data. The possible rights, namely *read* ( $R$ ), *write* ( $W$ ) and *read/write* ( $RW$ ), are represented by the set  $X$ . Finally, the sets  $Req_j$  and  $P_j$  represent the data requirements and attributes respectively, which have to be described in more detail. Thus, we introduce a model, which helps us to determine requirements for a single data, with respect to security and data management issues.

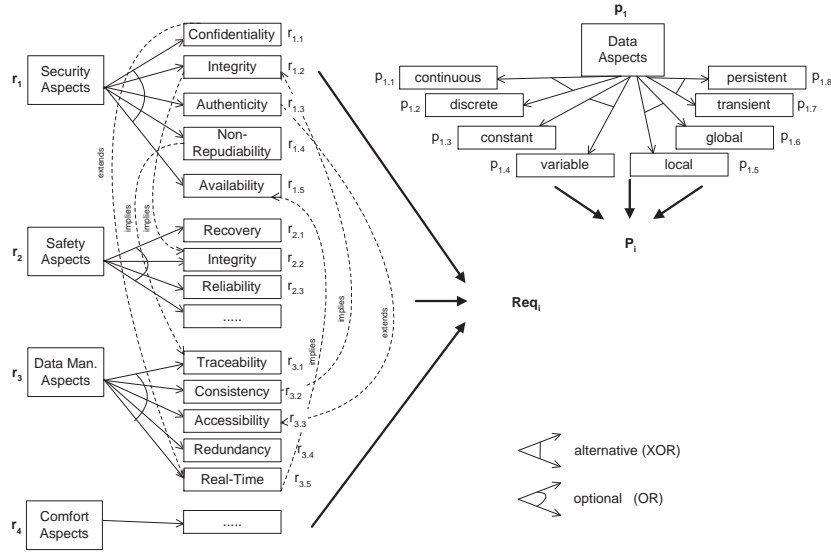
## 2.2 Modeling data requirements

Besides the formalisation, the model in Figure 2 is used for the modeling of data requirements. As can be seen, requirements are divided into different aspects, marked as  $r_1, r_2, r_3$  and  $r_4$ . Additionally, some *implicit* requirements are caused by data attributes, e.g., changeability, which are made explicit in the model as  $p_1$ . Furthermore, certain requirements depend on other requirements, which is denoted for *some* dependencies by *implies/extends*. Here, security and data management are integrated in our model. The specified requirements and attributes for a single data are represented by the sets  $Req_j$  and  $P_j$  respectively, which become part of the formal data definition (s. Section 2.1).

Considering all data (and its requirements) accumulated at one single ECU, a design guideline of the data management system for this ECU may be derived in two ways. First, requirements can be mapped directly to functional features, e.g. traceability can be mapped to a *logging* feature. Second, several requirements can be mapped to functional features using boolean operators. For instance, the requirements *continuous data flow*, *real-time* and *reliability* concatenated by the *AND* operator, possibly lead to a *main memory storage* feature, and thus, satisfy the integration aspect yet again. While the resulting features can be seen primarily as recommendations for the design phase so far, we plan to extend our model in that the features support the implementation phase in a straightforward way as well.

## 3 Applying the model to automotive data

One example for the application of the proposed model is the odometer reading. This data is highly sensitive to malicious alteration (i.e. a violation of IT security



**Fig. 2.** A Model for data requirements in automotive systems

aspects integrity, authenticity, non-repudiability, see also [4]). A successful attack on the odometer reading could result in a higher resale value of the car and thus, the owner or a dealer willing to sell the car could be interested in manipulating the counter value. As a result, an alteration of the odometer reading can have implications on both the *comfort* and the *safety* of the vehicle. For instance, the onboard computer computes the expected cruising range based on the odometer reading. Since there is a fuel gauge, alarming the driver of a low fuel reading, the consequences of such a miscalculation will not have safety consequences but will be a nuisance. However, a successful malicious alteration (i.e. violation of IT *security*) can also have consequences on the *safety* insofar as important maintenance measures with a specific service interval are not carried out, and thus, may be affect the powertrain-, brakes- and steering systems.

Applying our proposed model to the scenario described above (with respect to the topology mentioned in Section 2.1), leads to the exemplary set of ECU's  $C = \{c_1, c_2, c_3, c_4, g_1\}$ .  $C$  is used with the following exemplary configuration:

- $c_1$  dashboard
- $c_2$  onboard computer
- $c_3$  central lock
- $c_4$  engine control unit
- $s_7$  velocity sensor
- $g_1$  gateway between powertrain-, body electronics- and instrumentation-bus

With the given configuration of the system, the *odometer count* data, chosen as an example, can be modeled as follows:

$$\begin{aligned}
 d_1 &= \{\textit{odometer\_count}, C_1, Req_1, P_1\} \\
 C_1 &= \{c_4|\{W\}, c_1|\{R\}, c_2|\{R\}\} \\
 Req_1 &= \{r_{1.2}, r_{1.3}, r_{1.4}, r_{1.5}, r_{2.1}, r_{3.2}, r_{3.4}, r_{4.1}\} \\
 P_1 &= \{p_{1.2}, p_{1.4}, p_{1.6}, p_{1.8}\}
 \end{aligned}$$

The whole data is described by the formula  $d_1$ . As an unique identifier *odometer\_count* is chosen. While three ECUs are affected by the data ( $C_1$ ), the engine control unit  $c_4$  is the only device which has write access to this data. Other devices like the dashboard ( $c_1$ ) and the on-board computer ( $c_2$ ) will only have read access, which will be enforced by the DBMS components of the final system. Important requirements for this piece of data are represented in the set  $Req_1$ , e.g., the security target integrity ( $r_{1.2}$ ) is a central aspect. While processing the proposed model, the algorithm will demand an appropriate security algorithm to protect the integrity (e.g. digital signatures), ideally captured by the DMS (e.g. within a data validation module). Since the security target integrity ( $r_{1.2}$ ) already implies the related safety aspect integrity ( $r_{2.2}$ , e.g. implemented by CRC checksums), the latter does not have to be specified in the formula by the engineer anymore. In other words, integrity from the perspective of safety can be seen as a subset of the integrity from the IT security perspective. Finally,  $P_1$  lists important characteristics of this data, e.g. that it is a *global* ( $p_{1.6}$ ) *variable* ( $p_{1.4}$ ), which has to be stored *persistently* ( $p_{1.8}$ ).

Taking the result of the example model above, we can now derive some design recommendations in form of (abstract) functional features for the data management of *every* participating ECU. For instance, considering the dashboard ( $c_1$ ), which only has read access to the *odometer\_count* data. Possible functional features for this ECU are listed in Table 1 below.

Requirements/Attributes	Design recommendation
Authenticity, Accessibility	Access management feature with sophisticated authenticity validation, e.g. cryptographic algorithm
Integrity, (Redundancy)	Update feature using digital signatures
Consistency, Redundancy	Concurrency control feature
Non-Repudiability (implies Traceability)	Logging feature

**Table 1.** Exemplary relation of requirements and functional modules

It has to be mentioned, that the recommended features in Table 1 do not claim to be complete, regarding the underlying example. Moreover, we have chosen a quite simple example with only one piece of data considered here. Usually, a control unit has to handle a plenty of data, leading to an increasing number of requirements (and their combinations). Hence, the amount of possible functional features may be increase as well.

## 4 Conclusions and Future Work

In this paper, we have proposed the integration of IT security and data management, an issue, which was mostly neglected in its definition and detailed analysis in the automotive domain so far. Therefore, we presented a model where data requirements can be specified regarding different aspects and thus, integrate

IT-security and data management with 'classic' disciplines like safety and comfort. In addition, with the suggested model it is possible to derive functional features for a secure data management, even though this happens in a quite abstract way.

As future work, we want to combine our model with a feature diagram, representing the superset of all functionality relevant to automotive data management. This should help to make the relation between requirements and functional features more obvious. Additionally, we intend an implementation of this model, to proof its applicability.

Another future task is to evaluate the impact of single IT-components on the entire system with respect to both, function and structure in terms of IT-security and safety, in more detail.

Furthermore we intend to make appropriate IT security mechanisms for several requirements more explicit and take the data flow and its incorporated ECU's into account by using already existing models.

Finally, we want to examine, where to place this model within the AUTOSAR Methodology, as far as possible.

## 5 Acknowledgements

The work described in this paper has been supported in part by the European Commission through the EFRE Programme "COMO" under Contract No. C(2007)5254. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

## References

1. Pretschner, A., Broy, M., Krüger, I., Stauner, T.: Software Engineering for Automotive Systems: A Roadmap. In: Proceedings of Future of Software Engineering (FOSE) at the ICSE. (2007) 55–71
2. Casparsson, L., Rajnak, A., Tindell, K., Malmberg, P.: Volcano - a Revolution in On-Board Communications. Technical report, Volvo Technologies (1998)
3. Hoppe, T., Kiltz, S., Lang, A., Dittmann, J.: Exemplary Automotive Attack Scenarios: Trojan horses for Electronic Throttle Control System (ETC) and replay attacks on the power window system. In: Proceedings of the 23. VDI/VW Gemeinschaftstagung Automotive Security. (2007) 165–183
4. Lang, A., Dittmann, J., Kiltz, S., Hoppe, T.: Future Perspectives: The Car and its IP-Address - A Potential Safety and Security Risk Assessment. In: Proceedings of the 26th International Conference on Computer Safety, Reliability and Security (SAFECOMP). (2007) 40–53
5. Nyström, D., Tesanovic, A., Norström, C., Hansson, J.: COMET: A Component-Based Real-Time Database for Automotive Systems. In: Proceedings of the Workshop on Software Engineering for Automotive Systems at the ICSE. (2004)
6. Bishop, M.: Introduction to Computer Security. Number ISBN: 0-321-24744-2. Addison-Wesley (2005)