Systematic Elicitation of Mode Models for Multifunctional Systems

Andreas Vogelsang, Henning Femmer
Institut für Informatik
Technische Universität München, Germany
{vogelsan,femmer}@in.tum.de

Christian Winkler
MAN Truck & Bus AG
Engineering E/E System Vehicle Dynamic Functions (EEV)
München, Germany
christian.winkler.b@man.eu

***Abstract*—Many requirements engineering approaches structure and specify requirements based on the notion of modes or system states. The set of all modes is usually considered as the mode model of a system or problem domain.**

However, it is neither clear how such a mode model can be elicited systematically, nor whether it is realistic to elicit a mode model for a productive system with regard to size and comprehensibility.

In this paper, we introduce three elicitation approaches for mode models. We applied the three approaches in an industrial automotive context and assessed the resulting mode models with respect to size, complexity, and differences to each other.

Our results show that all elicitation approaches were capable of eliciting modes, which were structured in mode models with 20 to 42 modes. From these results, we conclude that it is possible to elicit manageable mode models for an entire system in a productive context. In our case, the practitioners decided to integrate our model in their feature specification and analysis process.

***Index Terms*—Requirements modeling, feature specifications, statecharts, industry, automotive software**

I. Introduction

The IEEE standard 29148 for software requirements specifications (SRS) denotes: "Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of features depending on its mode: training, normal, or emergency" [1]. The SRS proposes a specification structure for systems with modes in its annex. The use of statecharts [2] or other state-based specification techniques inherently relies on the advantages of structuring a system or a problem domain according to (observable) states. Use cases or scenarios may reference modes as trigger, pre- or postcondition [3]. We speak of modes when talking about a system's state of operation.

Systems that offer a variety of different features to their environment are called multifunctional systems [4]. The features of these systems serve different purposes and behave independently to some extent. However, the features of a multifunctional system can have subtle dependencies and may affect each other in certain situations (*feature interaction*) [5]. Along with the large number of features, these systems are also characterized by a large number of modes that influence their behavior.

We call the set of all modes that are used to formulate requirements the *mode model* of a system or a problem domain in general. Modes can be used explicitly (e.g., the mode `Engine Off` in a statechart) or implicitly (e.g., the term *engine is running* in the informal requirement "While the engine is running, the cooling control must maintain the motor temperature to a constant level").

Problem Statement: While many specification techniques rely on the notion of modes or states, there are no systematic approaches for eliciting a set (a model) of modes for a system. This could be especially problematic in the context of multifunctional systems [4], which may have a large number of modes to consider.

Contribution: In this article, we introduce three elicitation approaches for modes of a multifunctional system. One is based on expert interviews, one on an automated dependency analysis, and one on a manual requirements inspection. In an exploratory study, we applied these three approaches in the context of the development of a truck at MAN Truck & Bus AG and assessed whether the approaches are feasible, the resulting mode models are manageable in terms of their size and complexity, and how the qualitative characteristics of the resulting modes differ for the elicitation approaches. We were able to elicit three mode models with 20 to 42 modes and created an integrated reference mode model for the examined context with an overall of 75 modes.

Context: We conducted the study described in this paper at MAN Truck & Bus AG, an international manufacturer of commercial vehicles and transport systems because automotive systems are a good example for multifunctional systems as they provide features ranging from multimedia applications over driver assistance features to diagnosis features. The company has over 34,000 employees worldwide of which 150 work on electronics and software development. The organization's development process is supported by an integrated data backbone developed on the eASEE framework from Vector Consulting GmbH.

Outline: The remainder of the paper is structured as follows: In the next section, we list related work on state-based requirements specification and analysis techniques. In Section III, we give a detailed description of modes and mode models and introduce three elicitation approaches. In Section IV, we report on the design of a study, in which we apply the three elicitation approaches in an automotive context. After that, we report on the results of this study and discuss the threats to validity in Sections V and VI. Motivated by the results of our study, we introduce the development of a reference mode model for MAN in Section VII before we conclude the paper in Section VIII.

II. Related Work

The idea of modeling requirements by means of modes is described in many approaches (e.g., [4], [6], [7], [8]).

Broy [4] proposes to capture the requirements for a multifunctional system as services and to structure them in a service hierarchy. He specifies the services based on a formal system model, in which a service is described as an input processing and output producing function. Functional dependencies between services are described by inter-service communication, which the author calls mode channels. The hypothesis behind this is that inter-service communication can always be described by a mode (e.g., "If the car is driving faster than 20 km/h, the video player should be switched off."). However, a systematic elicitation approach for these modes is missing.

The situation is similar for the Software Cost Reduction Method (SCR) [7]. In SCR, *mode classes*, whose values are called *modes*, are used as auxiliary variables for the concise specification of required system behavior. A systematic approach on eliciting these mode classes is also not given for SCR.

Dietrich and Atlee [6] provide a generic structure for a feature into three high-level states `Inactive`, `Active`, and `Failed`. From our experience with features at MAN, this basic structure is too general in most cases. Almost all states need to be defined in more detail to provide an expressive model of the general solution concept. During an `Active` state, for example, a feature may run through a number of additional states or phases. Additionally, a feature might have a number of different degraded behaviors depending on the failure detected. The approaches presented in this paper result in much more fine-grained models of system modes. The characterization of feature behavior by states is also the basis for the feature-oriented requirements modeling language (FORML) [8]. In this language, states of a feature may be used to model feature interaction by referencing them in strengthening clauses of the guards of another feature's state transitions.

Well-known safety analyses also rely on the notion of system states. Fault Tree Analysis (FTA), for example, is a top-down deductive safety analysis technique. It is primarily a means for analyzing causes of hazards [9]. In an FTA, an analyst assumes a particular system state and a top event and then writes down the causal events related to the top event and the logical relations between them. The leafs of the tree often correspond to states of components of the system. Thus, an FTA also results in a state model, however, only from the point of view of states causing a hazard. Failure Modes and Effect Analysis (FMEA) is an inductive analysis method, which allows us to systematically study the causes of component faults, their effects, and means to cope with these faults. FMEA is used to assess the effects of each failure mode of a component on various functions of the system as well as to identify the failure modes significantly affecting dependability of the system. FMEA supplies the information about failure modes of the individual components into the FTA. FTA and FMEA are often conducted together to complement each other [10].

In summary, even though these approaches are all based on the modes of a system, none of the works proposes ways for their systematic elicitation. This is critical because an incomplete, inconsistent, or imprecise mode model may invalidate the results of a state-based specification or analysis approach. Moreover, the stated specification techniques refer to modes only as auxiliary means to provide a concise specification. In this paper, we show that a mode model by itself is a valuable entity for understanding and specifying a system.

III. Mode Models and Elicitation Approaches

A *mode* is a specific state that describes a system's state of operation. We describe a mode by a name and a set of *mode values* (e.g., $\texttt{Operation} = \{\texttt{Off}, \texttt{Starting}, \texttt{Running}\}$). A *mode model* is a (structured) set of modes of a system or a problem domain. Fig. 1 shows an excerpt of an exemplary mode model, including a typical hierarchical structure. The mode `Operation`, for example, has mode values `Off`, `Starting`, and `Running`, while `Running` is a mode for itself with mode values `Idle running active` and `Idle running not active`. That means if the engine is in operation mode `Running`, then the `Idle running` is either active or not active. Therefore, `Idle running active` and `Idle running not active` are *alternative* modes (at each point in time either one or the other is true). In contrast, the modes `Operation` and `Ignition` are *parallel* modes because `Engine` as their parent mode is characterized by a combination of mode values for both modes.

We use statecharts as introduced by Harel [2] and included in the UML [11] to represent mode models because AND and XOR decomposition of states in statecharts matches exactly the notion of parallel and alternative modes as introduced in the previous paragraph.

Some modes summarize a set of modes under a specific name. These modes that are exclusively composed of parallel modes are called *mode categories* (e.g., `Engine` is a mode category in the example).

We aim at exploring different approaches for eliciting modes of a multifunctional system. In this paper, we will investigate three approaches: (1) a top-down approach, where modes are defined based on the domain knowledge of experts, (2) a bottom-up approach, where modes are extracted from a given architecture of logical components and an analysis of dependencies between features within this architecture, and (3) an analytical approach, where requirements are manually inspected for implicit or explicit references to modes.

A. Elicitation by Interviews with Domain Experts

We assume that domain experts have a good intuition of what should be considered as a mode for a system based on their

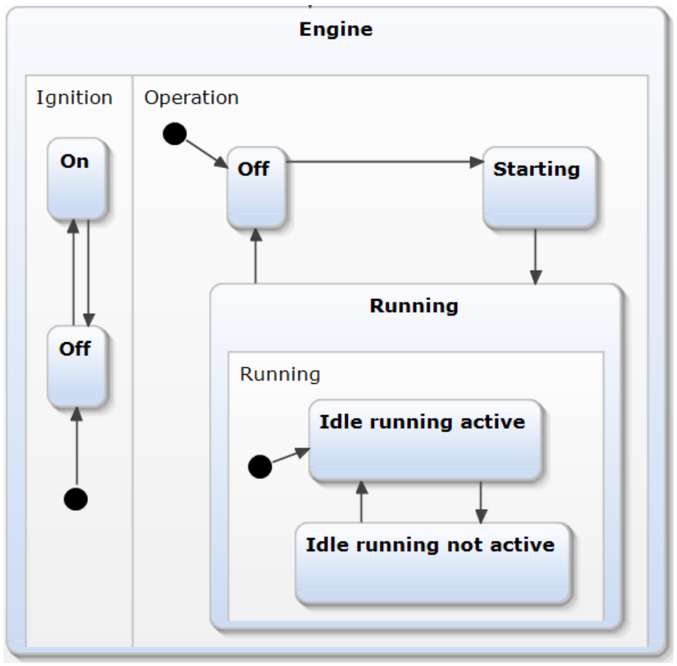

Fig. 1: Excerpt of an exemplary mode model represented by a statechart.

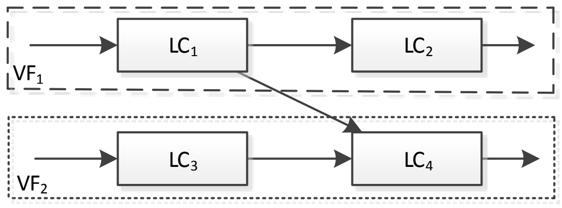

Fig. 2: The logical components (rectangles) are connected by data channels (black arrows) and form a functional architecture of the system. Vehicle features (VF) crosscut this architecture by the set of logical components that contribute to their realization (dashed forms). The data channel between $\texttt{LC}_1$ and $\texttt{LC}_4$ is a candidate for a mode.

experience in the development of features of a specific domain. Therefore, in this first elicitation approach, we obtain a mode model from interviewing domain experts. In the interviews, the interviewees are asked to enumerate what they consider as modes of the system under consideration. The result is a mode model that contains all modes mentioned in the interviews.

B. *Elicitation by Feature Dependency Analysis*

We define this second approach based on the results of a feature dependency analysis, because recent work, such as [4] and [6], postulates a relation between feature dependencies and modes of a system. In two former papers [12], [13], we presented the results of a feature dependency analysis performed on productive automotive systems. The analysis is based on a specific functional architecture of a system. In this functional architecture, logical components describe the realization/implementation of a vehicle feature in a purely logical fashion, i.e., without any information about the hardware the system runs on. A network of logical components describes the steps that are necessary to transform the input data into the desired output data. An example for a system that consists of 2 vehicle features (VF) that are realized by a network of 4 logical components (LC) is illustrated in Fig. 2. The logical components are afterwards deployed to a set of electronic computing units that execute the behavior of the logical components.

In the dependency analysis, a feature dependency is extracted from the functional architecture if there is a logical component associated with one feature exchanging data with a logical component associated with another feature (e.g., communication between $\texttt{LC}_1$ and $\texttt{LC}_4$). From this dependency analysis, we obtain a list of dependencies between features. Every item in this list contains two features (source and target) and the data signal that is responsible for the dependency.

Now, for this study, we assume all of these dependencies between VFs to be candidates for modes as postulated in [4], [6]: Each data signal represents a mode with the possible values of the data signal as mode values. A mode model is elicited by considering all dependency signals as modes and structuring them into mode categories.

C. *Elicitation by Requirements Inspection*

This third approach is based on natural language requirements, which are documented in requirements specifications. These specifications often implicitly contain statements about modes or states of a system. For example, the requirement “The air conditioning must maintain the desired temperature if the engine is running.” refers to a mode of the engine, namely that the engine is in mode `Running`. If a requirement refers to a mode value, we extract this mode value in this elicitation approach. Afterwards, the extracted mode values are grouped into modes (e.g., mode values `Running` and `Off` are grouped to mode `Engine Operation`). For the sake of clarity, the modes are finally structured into mode categories.

IV. Study Design

Based on the introduced elicitation approaches, we conduct a study, in which we apply the approaches in an industrial context.

A. *Research Objective*

Our study aims at exploring the different elicitation approaches for mode models of a multifunctional system and assessing the resulting models in terms of their size and complexity and qualitative differences to each other in the context of multifunctional systems.

B. Research Questions

RQ1: Are the introduced elicitation approaches feasible in practice?
First, we analyze the elicitation activities: We are interested whether the proposed elicitation approaches are capable of extracting a (non-trivial) mode model for a multifunctional system in practice.

RQ2: Are the resulting mode models manageable?
Second, we inspect the size and complexity of the resulting models: It is an open question, how large and how complex a mode model can get for a realistic system.

RQ3: How do the elicited modes differ?
Last, we qualitatively inspect the contents of the models: Different elicitation approaches may result in different modes and mode models. In this study, we want to explore the impact of the elicitation approach on the characteristics of the resulting modes.

C. Study Object

To answer the research questions, we applied the elicitation approaches in the context of the development of a truck at MAN Truck & Bus AG. Although the study objects for the three elicitation approaches were similar, they were not exactly the same. For the elicitation by interviews, we asked for modes relevant to the developer's domain of expertise and not for a specific vehicle. In the context of our study, the developers were usually responsible for the development of a number of features related to one domain (e.g., driver assistance). However, not all developed vehicles exhibit the full set of features. For the elicitation by feature dependency analysis, we analyzed the functional architecture of a compact truck with a restricted set of 55 features, while for the elicitation by requirements inspection we examined requirements of a fully equipped heavy truck. This difference in the study objects was due to the availability of data at the time of the study execution. The threats that this poses to the validity of the results are discussed later.

D. Data Collection Procedures

We elicited a mode model for the analyzed study objects by applying the three elicitation approaches introduced in Section III. Each of the elicitation approaches resulted in a mode model that was afterwards analyzed to answer the research questions. In the following, we report on the details of how we applied each elicitation approach in the context of our study.

1) Elicitation by Interviews with Domain Experts: We conducted interviews with four developers from the domains cabin & lights, base software, energy management, and driver assistance. These interview partners were selected by the head of the company's architecture group with the expectation to provide the maximum number of modes. Each interview lasted one hour with two interviewers and one interviewee participating. The stated modes were simultaneously written down by the interviewers and afterwards validated with the interviewee up to a point, where the interviewee considered the recorded mode model as representative for her domain. The interviews were not structured by any questionnaire or guideline. However, we guided the interviewee by asking open questions regarding specific classes of modes (e.g., "Are there any modes characterizing the vehicle's surroundings?"). After the four interviews, we documented the recorded modes in one integrated mode model containing all modes mentioned in the interviews. For this purpose, we merged equal modes mentioned in different interviews to one mode of the final integrated model. We considered two modes from different interviews to be equal if they have a similar name and at least one similar mode value. If, for example, in one interview the mode $\texttt{Engine} = \{\texttt{On}, \texttt{Off}\}$ was mentioned and in another interview the mode $\texttt{Engine Operation} = \{\texttt{On}, \texttt{Initializing}, \texttt{Off}\}$ was mentioned, we considered them as one mode mentioned in both interviews and added a merged version of that mode to the final integrated mode model.

2) Elicitation by Feature Dependency Analysis: In a former paper [12], we presented the results of a feature dependency analysis for a compact truck with a restricted set of 55 features. This analysis was performed completely automated by an analysis tool. From this dependency analysis, we extracted an overall of 91 different data signals responsible for all feature dependencies. Each data signal is additionally characterized by a data type within the company's data backbone. Based on these data types, we derived a mode model from this list of data signals in the following way. For each signal, we checked the data type:

Boolean Signals: If the data signal had a Boolean data type, we defined a mode with the name of the signal and associated mode values `Yes` and `No` (e.g., $\texttt{BrakePedalPressed} = \{\texttt{Yes}, \texttt{No}\}$).

Enumeration Signals: If the data signal had an enumeration as data type, we defined a mode with the name of the signal and the enumeration members as associated mode values (e.g., $\texttt{TransmissionMode} = \{\texttt{Park}, \texttt{Neutral}, \texttt{Drive}, \texttt{Reverse}\}$).

Value Signals: If the data signal had a numeric data type such as km/h, we tried to investigate how the signal is used. Since we were not able to access the actual code, we scanned the requirements specification of the corresponding vehicle feature in order to find a *discretization* of the data signal. For example, in the requirement specification of one feature, the signal `VehicleSpeed` was only used to distinguish between *low speed* and *high speed*. If we found a discretization of a numerical signal in the requirement specification, we defined a mode with the name of the signal and the extracted discrete values as associated mode values (e.g., `VehicleSpeed={Low,High}`).

If we were not able to transform a data signal into a mode by one of the three ways, we excluded this signal from the study. Similar to the proceeding for the modes of the interviews, we documented the resulting modes of the feature dependency analysis in one mode model.

3) Elicitation by Requirements Inspection: For eliciting modes from their implicit usage within requirements, we inspected 223 requirements from a randomly picked sample of 11 vehicle features of a heavy truck with an overall of about 150 features. A requirement in our study is a textual statement consisting of 1–2 sentences in general. The inspection was performed simultaneously by two researchers, who stepped through the requirements of one feature and extracted terms that matched a set of criteria as modes. In consistency with the other two approaches, these criteria were:

- A mode may change its value during runtime of the system (this excludes, for example, configurable parameters)
- A mode has a discrete set of mode values (this excludes continuous signals)
- A mode must be observable from outside the system (this excludes internal states)
- A mode's granularity must maintain a specific notion of importance for the entire system (this excludes, for example, specific failure states that are only relevant to one specific feature).

Similar to the proceeding for the modes of the other two elicitation approaches, we documented the resulting modes of the requirements inspection in one mode model.

E. Data Analysis Procedures

To characterize, assess, and compare the resulting mode models, we collected a number of quantitative measures,l which will be introduced in the following. Since our research questions cannot be answered by solely collecting quantitative measures, we use these measures as input for a qualitative feedback of MAN's head of the architecture department, who subjectively answered the research questions applying his domain knowledge and experience.

To indicate complexity, we collected three measures: Number of modes (NM), number of mode values (NMV), and nesting depth (ND). These measures support the assessment whether an elicitation approach is considered as feasible (RQ1) and the resulting mode models are considered as manageable (RQ2).

Number of Modes (NM): As a first measure, we counted the number of modes. Submodes, i.e., modes that are also mode values for another mode, were counted separately. The mode model shown in Fig. 1, for example, has three modes (`Ignition`, `Operation`, and `Running`).

Number of Mode Values (NMV): As a second measure, we determined for each mode model the number of mode values per mode. In the example of Fig. 1, the modes `Ignition` and `Running` each have two mode values, while `Operation` has three.

Nesting Depth (ND): As a third measure, we determined the nesting depth of each mode in a mode model. The nesting depth is determined by the length of the path through the hierarchical mode model to the mode. In the example of Fig. 1, mode `Ignition` has nesting depth 2 (there is an additional *root* node in the mode model). For both, number of mode values and nesting depth, we show the distribution by a box plot diagram including median, maximum, and minimum values.

For a comparison of the resulting mode models (RQ3), we explored the characteristics of the elicited modes from the three mode models. As an important facet for the characteristic of a mode, we classify a mode with respect to the *scope* it addresses.

Scope of Modes: The scope of a mode denotes the object of which a mode describes a property. To investigate and quantify this characterization in more detail, we classified the elicited modes of the mode models into the following three scopes:

Context: Modes with this scope describe states of the operational environment of the system under consideration (e.g., $\texttt{AmbientTemperature} = \{\texttt{Low}, \texttt{High}\}$).

System: Modes with this scope describe states of the system under consideration (e.g., $\texttt{DrivingDirection} = \{\texttt{Forward}, \texttt{Backward}\}$).

Feature: Modes with this scope describe states of specific features of the system under consideration (e.g., $\texttt{CruiseControl} = \{\texttt{Off}, \texttt{Standby}, \texttt{Active}\}$).

With this classification, we can investigate the qualitative differences between the modes elicited in the three elicitation approaches. More fine-grained or other classifications of mode scopes are also possible (e.g., scope with respect to a domain of a car). We report on the distribution of modes with respect to this classification for each mode model and thereby compare the characteristics of each elicitation approach. The purpose behind this is that we are interested in classes of modes that are only elicited by a specific elicitation approach.

F. Validity Procedures

To ensure the validity of our results, we performed several validity procedures to mitigate mainly threats to the internal validity of the study.

One major point of discussion for the validity of the study design is the fact that we applied the elicitation approaches to slightly different study objects (see Section IV-C). Unfortunately, we could not avoid this due to the availability of data at the time of the study. However, we mitigated this threat through two means: First, where possible, we selected comparable features of the system for the analysis. For example, for elicitation by requirements inspection, we analyzed a subset of the features for the dependency analysis. Second, we carefully avoided interpretations that could have been flawed due to this threat: For example, since the size of the resulting model might result from variance in the study object, we chose to compare the results qualitatively instead of quantitatively.

The elicitation of a mode model by interviews poses the threat that the mentioned modes were incorrectly documented by the researchers. This could be due to imprecise or incorrect interpretation of what was being said or even by a bias of the researcher in any direction. To mitigate this threat, we took notes in the presence of the interviewees letting them intervene in case they found something was noted incorrectly or imprecise.

The extraction of a mode model based on a static analysis of feature dependencies poses the threat that this automated analysis delivers incorrect or incomplete results that may then corrupt the resulting mode model. To mitigate this threat, we selected the same study object that was subject to a former project that especially focused on this automated dependency analysis. In the context of this former project, the dependency analysis results were extensively reviewed and published [12]. The extraction of the mode values for the dependency signals as described in Section IV-D2 were taken from the company's database and are thus not subject to a researcher bias or interpretation.

The elicitation of a mode model by requirements inspection poses the threat that the extraction of modes is subject to the researcher's subjectivity. To mitigate this threat, we inspected a set of 50 requirements independently by two authors of this paper, classifying whether or not a requirement contains a mode. For this set, we observed an inter-rater agreement in terms of Cohen's kappa of 0.63 (*substantial agreement* [14]). We considered this as an indicator that the authors had a good agreement on what should be considered a mode. We resolved these deviations in a discussion and fixed the decision points in a set of strict criteria that defined what is considered as mode in this study (see Section IV-D3). These criteria were then taken as a basis for further inspection of requirements. From this, we conclude that the extracted mode model is fairly reliable considering researcher's subjectivity.

The classification of modes with respect to their scope also poses the threat that the classification is subject to the researcher's subjectivity. To mitigate this threat, we performed the classification by two researchers. A first round of independent classification resulted in an inter rater agreement in terms of Cohen's Kappa of 0.54 (*moderate agreement* [14]). Deviations in the classification only appeared between the *system* and the *feature* scope. We resolved these deviations in a discussion.

Another threat that may especially affect results of the manual elicitation approaches (interviews and inspection) is the order of executing the elicitation. To mitigate that the authors learned about modes during the interviews, which they then tried to identify in the requirements or vice versa, we performed interviews and inspection sessions interleaved (i.e., not in a specific order).

V. Study Results

In the first part of this section, we report on the quantitative measures of the elicited mode models for the three elicitation approaches. In the second part, we discuss the research questions based on a qualitative assessment with the head of MAN's architecture department.

A. Elicited Mode Models

We were able to elicit non-trivial mode models by all of the three elicitation approaches, i.e., none of the approaches was a dead end.

Table I shows an overview of the quantitative measures that we collected for the three elicited mode models.[1] Be aware that the absolute numbers may not be comparable due to the different study objects. We only compare the distribution of scopes within the mode models to each other.

[1]The additional *Reference* plot can be ignored for now and will be discussed in Section VII

1) Mode Model from Interviews: The mode model that we assembled from all modes of the interviews contains 42 modes (*NM*). The elicited modes have 2 mode values on average (*NMV median*) with a maximum of 6 mode values for one mode. The average nesting depth (*ND median*) of the modes in the resulting mode model is 2 with a maximum nesting depth of 4. Table II lists the number of elicited modes for each interview and the ratio of modes that were also mentioned in at least one of the other interviews (*common modes*). As shown in the table, the number of modes elicited from the interviews range from 7 to 24 and the ratio of modes that were also mentioned in another interview ranged from 27% to 71%. 12/42 modes (28%) of the final mode model were mentioned in more than one interview.

The modes elicited based on interviews are widely distributed over all classes of scopes (*Context*: 14%, *System*: 50%, *Feature*: 36%). In comparison with the other approaches, the elicitation by interviews shows the lowest ratio of modes with a system scope, although the difference between the approaches in this scope category is only 5%. Interestingly, modes of the electronic infrastructure, such as the state of an electronic control unit or the activation of a bus system, were only elicited in the interviews but not in any of the other approaches. The slightly higher nesting depth of the mode model elicited from the interviews compared with the other approaches may indicate that it might be easier in this approach to determine structural relations between modes. This may be due to the fact that this approach is the only approach that exploits domain knowledge of experts, who can also share knowledge about structural relations between modes (e.g., one mode is a submode of another).

2) Mode Model from Feature Dependency Analysis: From the feature dependency analysis, we were able to elicit a mode model with 20 modes (*NM*). The extracted modes have 2 mode values on average (*NMV median*) with a maximum of 14 mode values for one mode. The average nesting depth (*ND median*) of the modes in the resulting mode model is 2 with a maximum nesting depth of 3. We were able to transform 35/91 (38%) of the dependency signals from the analysis into the 20 modes. The fact that there are less modes than transformed signals is due to signals that carried redundant information with respect to the mode (e.g., the signals `PedalPressed` and `PedalPosition` are both determined by the same mode `PedalState`). The dependency signals that could not be transformed into modes were numeric data for which we have not found any discretization in the requirements documents.

The mode model resulting from the dependency analysis did not contain any context modes (*Context*: 0%). A reason for this might be that, with this approach only modes are elicited that originate from concrete data signals within the functional architecture of a system (cf. Fig. 2). Modes with a context scope are typically rather abstract and general (e.g., `Bad weather condition` or `Presence of oncoming vehicle`). Therefore, they may never appear as signals in the implementation. The high number of dependency signals that could not be transferred into a mode

TABLE I: Overview of quantitative measures for the elicitation approaches.

		Elicitation approach							
		Interview		Dep. Analysis		Inspection		Reference	
	NM	42		20		39		75	
	NMV Median (Min–Max)	2	(2–6)	2	(2–14)	2	(2–7)	2	(2–16)
	ND Median (Min–Max)	2	(1–4)	2	(1–3)	2	(1–3)	2	(1–5)
Scope	Context	6	(14%)	0	(0%)	8	(21%)	11	(15%)
	System	21	(50%)	11	(55%)	21	(54%)	36	(48%)
	Feature	15	(36%)	9	(45%)	10	(26%)	28	(37%)

TABLE II: Number of modes elicited in the interviews and ratio of modes that were mentioned in more than one interview

Domain	**Modes**	**Common Modes**
cabin & lights	24	12 (50%)
base software	13	5 (38%)
driver assistance	11	3 (27%)
energy management	7	5 (71%)
Summary	**42**	**12 (28%)**

(see Section V-A2) are an indicator that a large portion of inter-feature dependencies on an implementation/architecture level are due to architectural or technical decisions and do not reflect dependencies on the level of functional requirements, for which we assume that they correspond to modes (see Section III-B). For example, the dependency analysis also contained priority signals for the order of computation of functional blocks. This mismatch between dependencies on an implementation and requirements level is also addressed by the *optional feature problem* [15]. Another interesting aspect of the elicited mode model is the high portion of modes associated with a feature scope (*Feature*: 45%; e.g., `ABS active`). These feature-oriented modes were not as extensively elicited in the other approaches as in this approach. This is even more interesting since in the dependency analysis, only modes are elicited that originate from a dependency between features of a system. Modes that are only relevant in one feature cannot be elicited with this approach since they are not detected by the dependency analysis. Therefore, this result supports the hypothesis that modes with a feature scope play an important role for the description of dependencies between features (cf. [4], [6]). What is striking for the modes originating from the dependency analysis is their detailed character. These modes describe, for example, specific pressing scenarios for a pedal (e.g., `Long-press` vs. `Short-press`). Together with the fact that the number of mode values per mode in this approach is higher than in the other approaches indicates that this elicitation approach results in more detailed and more fine-grained modes.

3) Mode Model from Requirements Inspection: In the inspection of requirements, we observed that 165/223 (74%) of the requirements contained information regarding at least one mode. From the analyzed requirements, we extracted 39 modes (*NM*). The extracted modes from the requirements inspection have 2 mode values on average (*NMV median*) with a maximum of 7 mode values for one mode. The average nesting depth (*ND median*) of the modes in the resulting mode model is 2 with a maximum nesting depth of 3.

The elicitation of modes based on requirements inspection elicited 39 modes distributed over all scopes (*Context*: 21%, *System*: 54%, *Feature*: 26%). Many of the modes specific to the requirements inspection elicitation were associated with a context scope and therefore described rather abstract and general states, such as the aforementioned `Bad weather condition`. This is similar to the modes of the elicitation by interviews and reflects that the requirements are written in the language of the domain experts. Such modes seem to be important for the understanding and documentation of functionality. It is interesting that we did not find corresponding signals of these modes in the implementation (cf. dependency analysis elicitation approach *Context* scope: 0%).

The box plots in Fig. 3 show a comparison of the number of mode values elicited for each mode and its nesting depth in the mode model of the different elicitation approaches.[2] The figure shows that the number of mode values for each mode tends to be slightly higher for the modes elicited by the dependency analysis, whereas for the other two approaches the number is similar. We cannot observe a striking difference in the nesting depth of the models; there is only a slight tendency for the interview model that the nesting depth is 2 at minimum.

B. Discussion of Research Questions

As stated before, we presented the resulting mode models and the collected measures in a workshop with MAN's head of the architecture department. In the context of this workshop, we qualitatively discussed and answered the research questions specifically for this case, making use of the domain expertise of the MAN head of architecture.

1) RQ1: Feasibility of the Approaches: In this research question, we wanted to assess whether the elicitation approaches result in models that are correct in the sense that they contain only modes that are also considered as modes by a domain expert. Additionally, we required that approaches do not produce trivial models, for which it might not be worth developing systematic elicitation approaches. Both points were confirmed by the domain expert of MAN. With a minimum

[2]The additional *Reference* plot can be ignored for now and will be discussed in Section VII

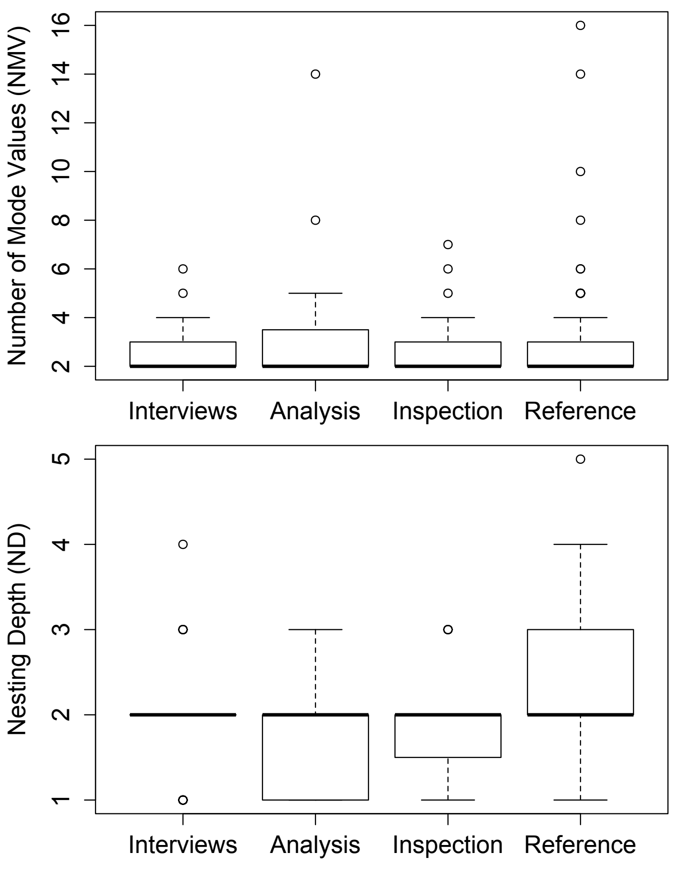

Fig. 3: Box plot for the distribution of the number of mode values (NMV) and nesting depth (ND).

number of 20 modes, he considered all model models as "not trivial". He additionally confirmed that he would consider all elicited modes as actually being modes, i.e., he assessed the mode models as being correct.

2) RQ2: Manageability of the Mode Models: A central aspect that we wanted to answer by this study is the question whether the elicitation of a mode model for an entire productive system results in a model consisting of hundreds or even thousands of modes that, in the end, is not understandable, usable, or maintainable anymore. By the results of our study, we are convinced that this is not the case. The mode models we elicited had 20 to 42 modes and the structures of the models were relatively flat (*nesting depth median* = 2). In the workshop, the domain expert from MAN confirmed this claim. From his experience, the function developers are capable of understanding and maintaining models of such a size and complexity. As a reference for comparison, he used the average number of logical components that one function developer needs to understand and maintained, which is roughly 30.

3) RQ3: Differences in the Mode Models: In summary, we were surprised by the diversity of the modes resulting from the different elicitation approaches. Although their characteristics, in terms of the scopes they address, were quite different, all modes have merit for describing a system's current state of operation. This was confirmed by the domain expert and therefore, we consider all of the approaches as beneficial and leading to reasonable results.

VI. Threats to Validity

Despite the applied validity procedures (see Section IV-F), the study design still poses some threats to the validity of the results: As described before, the study objects used for the elicitation approaches were not exactly the same. A consequence of this is that the absolute numbers of the elicited modes and size of the mode models are not comparable to each other. Therefore, we only assessed qualitative differences between the elicitation approaches because we are convinced that the differences in the study objects may have an impact on the number of elicited modes but not on their characteristics. The structure of the mode models was partly extracted by the researchers and thus not strictly determined by the elicitation approach itself. However, this fact may only influence the measured nesting depth. The low ratio of common modes for the elicited modes in the four interviews may suggest that additional interviews might lead to additional modes and thus the elicited mode model might not be complete. However, the interviewees were selected externally according to the head of the company's architecture department with the goal to cover a broad range of modes.

For all of the three elicited mode models we cannot guarantee completeness of the elicited mode models because we only conducted four interviews, only inspected requirements from 11 features, and only analyzed fully specified parts of the systems. Future work should investigate the completeness of the created models.

Lastly, we answered the research questions for our specific case study. Future work must be conducted to refute or confirm our answers in other contexts.

VII. Reference Mode Model

Based on the results of our study, MAN decided to use mode models in their feature specification and analysis process. Motivated by the diversity of the elicited mode models (40–60% of the modes of one approach were specific to that approach), they decided to use a combined mode model as a comprehensive model of the operational states of a vehicle system. Therefore, we created a *reference mode model* by merging the modes of all elicitation approaches. For this purpose, we identified equal modes of the three models and merged the mode values of the matching modes to a mode of the reference model. Similar to the proceeding for merging the modes from different interviews into one integrated model (see Section IV-D1), we consider two modes from different mode models to be equal if they have a similar name and at least one similar mode value. The modes that could not be matched were added unchanged to the reference model.

The Venn diagram in Fig. 4 shows how we assembled the reference mode model from the modes of the three elicitation approaches. The figure shows that 4 modes (5%) appeared in all elicited mode models. These modes are (1) the operation

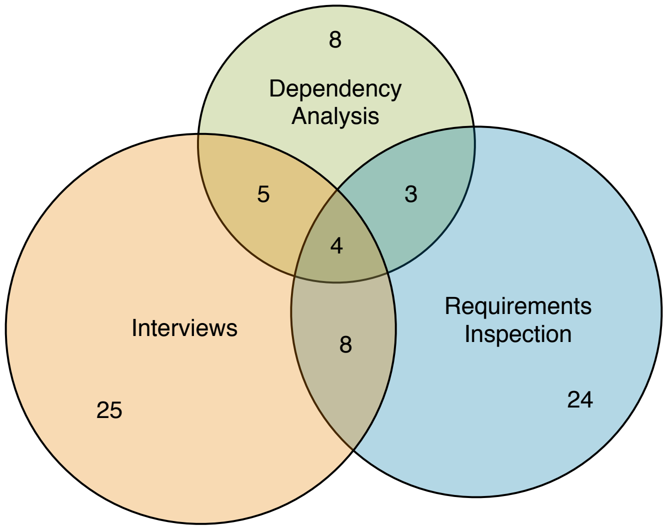

Fig. 4: Distribution of modes of the reference model with respect to the original three mode models.

status of the engine, (2) the information whether the drive train is released, (3) the information whether the acceleration pedal is pressed, and (4) the current cruising range. An overall of 57 (74%) of all modes originate from the mode model of just one elicitation approach.

In the end, the reference model contained 75 modes that we structured into mode categories similar to our proceeding for the three elicitation approaches. Table I and Fig. 3 shows the quantitative measures and the distribution of mode values (*NMV*) and nesting depth (*ND*) that we collected for this reference model.

The nesting depth tends to be higher than in the original models, which is not surprising, assuming that a larger number of modes suggest additional mode categories to maintain the understandability. The number of mode values per mode is similar to those of the original models; however, there is a larger number of outliers with significantly more mode values than in the original models. The classification of scopes with respect to their *scope* shows a distribution for the reference mode model of: *Context*: 15%, *System*: 48%, and *Feature*: 37%.

We consider this reference model as a comprehensive model of operation modes for the development of a truck at MAN Truck & Bus AG since it considers and contains all viewpoints addressed by the three elicitation approaches. The development of a new truck or the evolution of an existing truck may profit from this reference model as a starting point for formulating and modeling functional requirements.

VIII. Conclusions

In this paper, we presented three approaches for eliciting a mode model for a multifunctional system. The purpose of a mode model is to describe a system in terms of states of operation. Such a mode model may serve as a basis for a state-based specification of system functionality or safety aspects.

We applied the three elicitation approaches in the context of the development of a truck at MAN Truck & Bus AG. In our case study, we answered the research questions in the following way:

RQ1: Are the introduced elicitation approaches feasible in practice?
Yes, in our study all three elicitation approaches were capable of eliciting modes, thus all approaches resulted in a (non-trivial) mode model.

RQ2: Are the resulting mode models manageable?
Yes, the size of the elicited models ranged from 20 to 42 modes and even a combination of all mode models (*reference mode model*) did not exceed 75 modes.

RQ3: How do the elicited modes differ?
With all three approaches, we elicited modes that were solely elicited by one approach. We thereby conclude that none of the approaches was superfluous (e.g., if most of its modes were also elicited by another approach).

Based on the answers to RQ1 and RQ2, MAN's head of architecture decided to integrate mode models in MAN's vehicle feature specification and analysis process. Based on the answer of RQ3, MAN decided to use our developed reference model as the initial mode model for the development of new features and vehicles.

The aim of this study was to explore different elicitation approaches for mode models and assessing the resulting models. A subsequent question that was not in the focus of this study is which approach is applicable under which conditions and how these elicitation approaches should be integrated in a requirements engineering process. An integration of the elicitation approaches into a requirements engineering process must, for example, also consider the necessary effort. Conducting interviews is time and person intensive; however, their analysis and the extraction of a mode model can be performed rather quickly. In our study, conducting an interview took one hour and one additional hour for its analysis and mode extraction. The extraction of modes from requirements is even more time intensive but can be performed by a single (or a smaller group of) person(s). In our study, the inspection of one specification took approximately one hour. The extraction of modes by the dependency analysis can be performed automatically to a large extent, i.e., the dependency analysis is completely automated and the transition of data signals to modes must only be performed manually in cases where the signal's data type is a *value* type (see Section IV-D2).

Besides the practical implications in the context of MAN, our results also have a relation to existing work in the academic context. Filipovikj et al. [16] mention the use of high-level concepts, such as *shutdown* or *start-up*, in textual requirements as an impediment to their formalization. In their study, they report on the difficulties inherent to the process of transforming system requirements from their traditional written form into semi-formal notations by applying specification patterns [17]. They observed two problems related to the mentioned high-level concepts. First, none of the predefined scopes of the specification patterns capture the moment when the system is “in” some specific state (e.g., *start-up*), and secondly, such states belong to a higher abstraction level and are not properly specified, so their meaning is ambiguous. The authors conclude that such high-level concepts need to be disambiguated by an engineer. Post et al. [18] performed a similar study. The examples of requirements they present as difficult/impossible to formalize also contain references to states (e.g., “The drag torque and the activation torque depend on the operating state”). A mode model may mitigate these impediments to formalization by providing high-level concepts with a precise meaning.

We used statecharts as a precise notation technique to describe mode models by a hierarchy of parallel and alternative modes. Statecharts additionally allow the definition of mode transitions, which have not been tackled so far in our study. However, adding mode transitions may facilitate validation and verification activities but requires additional information that is not elicited by our approaches. More general, it might also be interesting to state invariants over the mode model (e.g., a specific mode combination must never occur). The specification of a mode model for an entire multifunctional system may also lead to a completely new way of describing, specifying, and developing functionality by complementing classical function-oriented specifications (e.g., in Matlab Simulink).

As a next step, we plan to conduct a case study in which we assess the impact of using the elicited reference mode model as a basis to specify a new feature, i.e., extend an existing system with an additional feature. We expect that the usage of the mode model contributes to the completeness of the resulting feature specification because specific combinations of modes remind the developer of functional corner cases, which are often missed. Additionally, we expect that the mode model contributes to an unambiguous interpretation of high-level concepts used in textual requirements.

References

[1] IEEE, “Systems and software engineering – life cycle processes – requirements engineering,” *ISO/IEC/IEEE 29148:2011(E)*, 2011.

[2] D. Harel, “Statecharts: a visual formalism for complex systems,” *Science of Computer Programming*, vol. 8, no. 3, 1987.

[3] A. Cockburn, *Writing Effective Use Cases*. Addison-Wesley Professional, 2001.

[4] M. Broy, “Multifunctional software systems: Structured modeling and specification of functional requirements,” *Science of Computer Programming*, vol. 75, no. 12, 2010.

[5] M. Calder, M. Kolberg, E. H. Magill, and S. Reiff-Marganiec, “Feature interaction: a critical review and considered forecast,” *Computer Networks*, vol. 41, no. 1, 2003.

[6] D. Dietrich and J. M. Atlee, “A mode-based pattern for feature requirements, and a generic feature interface,” in *RE’13*, 2013.

[7] C. L. Heitmeyer, J. Kirby, and B. G. Labaw, “The SCR Method for Formally Specifying, Verifying, and Validating Requirements: Tool Support,” in *ICSE’97*, 1997.

[8] P. Shaker, J. Atlee, and S. Wang, “A feature-oriented requirements modelling language,” in *Proceedings of the 20th IEEE International Requirements Engineering Conference (RE’12)*, 2012.

[9] N. G. Leveson, *Safeware: System Safety and Computers*. Addison-Wesley, 1995.

[10] E. Troubitsyna, “Elicitation and specification of safety requirements,” in *Third International Conference on Systems (ICONS’08)*, 2008.

[11] OMG, *OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1*, Object Management Group Std., Rev. 2.4.1, August 2011. [Online]. Available: http://www.omg.org/spec/UML/2.4.1

[12] A. Vogelsang, S. Teuchert, and J. Girard, “Extent and characteristics of dependencies between vehicle functions in automotive software systems,” in *MISE@ICSE*, 2012.

[13] A. Vogelsang and S. Fuhrmann, “Why feature dependencies challenge the requirements engineering of automotive systems: An empirical study,” in *21st IEEE International Requirements Engineering Conference (RE’13)*, 2013.

[14] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, vol. 33, no. 1, 1977.

[15] C. Kästner, S. Apel, S. S. ur Rahman, M. Rosenmüller, D. Batory, and G. Saake, “On the impact of the optional feature problem: Analysis and case studies,” in *SPLC’09*, 2009.

[16] P. Filipovikj, M. Nyberg, and G. Rodriguez-Navas, “Reassessing the pattern-based approach for formalizing requirements in the automotive domain,” in *RE’14*, 2014.

[17] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett, “Patterns in property specifications for finite-state verification,” in *Proceedings of the 21st International Conference on Software Engineering (ICSE’99)*, 1999.

[18] A. Post, I. Menzel, J. Hoenicke, and A. Podelski, “Automotive behavioral requirements expressed in a specification pattern system: A case study at BOSCH,” *Requirements Engineering*, vol. 17, no. 1, 2012.