Take Care of Your Modes!
An Investigation of Defects in Automotive Requirements

Andreas Vogelsang[1], Henning Femmer[1], and Christian Winkler[2]

[1] Technische Universität München, Germany
{vogelsan,femmer}@in.tum.de,
[2] MAN Truck & Bus AG
Engineering E/E System Vehicle Dynamic Functions (EEV)
Munich, Germany
christian.winkler.b@man.eu

Abstract. [**Context & motivation**] Requirements for automotive software systems are predominately documented in natural language and often serve as a basis for the following development process. Therefore, requirements artifact quality is important. Requirements often contain references to specific states of a system, which we call modes (e.g., "While the system is running, . . . "). [**Problem**] However, these references are often implicit and therefore, we suspect them as possible source for misunderstandings and ambiguities. [**Principal idea**] In this paper, we explore the relation between quality defects of natural language requirements and the description of modes within them. For this purpose, we investigate review findings of industrial requirements specifications and assess how many findings contain issues addressing a mode and which defect types are most affected by mode-related findings. [**Contribution**] Our preliminary results show that 46% of all considered review findings contain issues addressing a mode. Defect types in which modes played a major role were *completeness* and *unambiguity*. Based on these results, we argue that explicitly specifying modes prior to requirements formulation may increase the artifact quality of natural language requirements specifications.

Keywords: Requirements modeling, feature specifications, natural language requirements, automotive software, industry

1 Introduction

The behavior of automotive (software) systems often depends on information that represents states of the system or its surrounding environment. States may influence the activation/deactivation of vehicle features (e.g., `low battery`), determine a specific feature behavior (e.g., `ACC in follow-up`), or describe feature interaction (e.g., `feature X failed` ⇒ `feature Y degraded` [2]). We call these states of operation *modes*. We know from previous studies [6] that modes can be classified into three categories: They may describe states of the surrounding environment (e.g., `high temperature`), of the system itself (e.g., `ignition on`), or of a system feature (e.g., `ACC active`). We also know that modes play an especially important role for the specification of *multifunctional systems*, which are characterized by a variety of different functions integrated into one system [1].

However, natural language is still the most common way to describe requirements, which often contain implicit descriptions of modes. 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 `engine running`. In the past, we focused on researching practical applications of mode models. One of the benefits that we (together with our partners in industrial practice) assume is that explicit management of modes has the potential to improve the quality of natural language requirements specifications.

In this paper, we present an empirical investigation of defects in natural language requirements specifications of industrial automotive features. The goal of this study is to assess the relation between quality defects of natural language requirements and the description of modes within them. The results of the study show that 46% of all considered defects address modes. The defect types in which modes played a major role were *completeness* and *unambiguity*.

Our results indicate that explicitly specifying modes may increase completeness and unambiguity in natural language requirements specifications.

2 Study Design

2.1 Goal and Research Questions

The goal of this study is to understand the relation between quality defects of natural language requirements and implicit descriptions of modes within them. To accomplish the stated goal, we aim at finding out how many review findings address system modes and which defect types are most affected by mode-related review findings. From this goal definition, we derive our research questions:

RQ1: How many defects in NL requirements specifications mention modes? With this RQ, we want to assess the extent of modes as a source for defects in NL requirements specifications. The answer to this question indicates the relevance of considering modes for NL artifact quality.

RQ2: Which types of requirements defects are issued by mode-related findings? With this RQ, we want to understand how implicit descriptions of modes impact the artifact quality of NL requirements. The answer to this question lists the expected quality defects within NL requirements containing modes.

RQ3: Which types of problems categorize the mode-related findings? With this RQ, we want to identify and articulate problems that cause the quality defects issued in mode-related findings. The identified problems define requirements for approaches that try to increase the artifact quality.

2.2 Case and Subject Description

To answer the research questions, we investigated reviews of requirement specifications for vehicle functions at our partner MAN Truck & Bus AG. MAN applies a rigorous reviewing process to assure a high quality in their development artifacts. Natural language requirements specifications for vehicle functions are subject to an extensive manual review after every major change. In this process, a specification is reviewed independently by at least three experts followed by a review meeting, in which a moderator, the responsible requirements engineer, and the reviewers discuss the findings and decide how to address them. There are no guidelines how to review a specification. In particular, there is no specific focus on modes. Findings and decisions are documented in a review protocol.

For our study, we inspected two of these review protocols that originated from the review process of two vehicle functions, one from the *driver assistance* and one from the *cabin & lights* domain. The reviews contained 134 findings for an overall of 41 requirements contained in the specifications.

2.3 Data Collection and Analysis

In the data collection process, we prepared and classified the data (see Fig. 1):

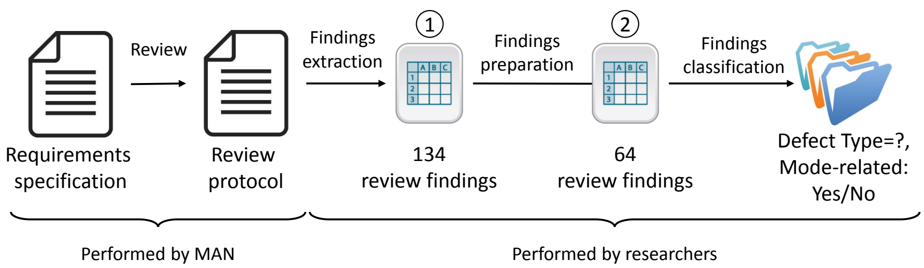

Fig. 1. Data collection process

Findings preparation: In the review process of MAN, review findings are classified as *major defect*, *minor defect*, *spelling mistake*, *question*, or *process improvement*. We reduced the initial set of 134 review findings (see ① in Fig. 1) by removing findings marked as *process improvement* because we are only interested in issues related to artifact quality, and we removed findings marked as *spelling mistake*. Furthermore, we removed findings that were marked as *rejected* in the review meeting because we are interested in issues that actually caused efforts to fix them. As a final preparation step, we reduced the data set by removing *duplicate* findings, i.e., issues that were mentioned by more than one reviewer. We removed these to ensure that the same finding is not counted multiple times. After this findings preparation, we ended up with an overall of 64 findings, which served as the basis for our classification (see ② in Fig. 1).
Findings classification: We classified each finding according to the following classification schemes:

1. **Defect type:** We used the defect type taxonomy of IEEE standard 29148 [3] to assign a defect type to each review finding (*complete*, *unambiguous*, *singular*, ... ; see [3] for details).
2. **Mode-related:** We used the definition of mode from [6] to decide whether or not a review finding mentions an issue related to a mode (*yes*/*no*).

To answer RQ1, we compared the number of all findings with the number of findings that we classified as mode-related. To answer RQ2, we ranked the defect types according to the number of mode-related findings in that defect type. To answer RQ3, we walked through the mode-related findings several times and annotated a keyword to each finding that describes the problem issued in the finding with respect to its content. As a result of this inductive categorization process, a set of problem classes emerged.

3 Study Results

Fig. 2 shows the distribution of review findings with respect to defect types. An additional column is attached to each defect type that shows the number of mode-related findings with that defect type.

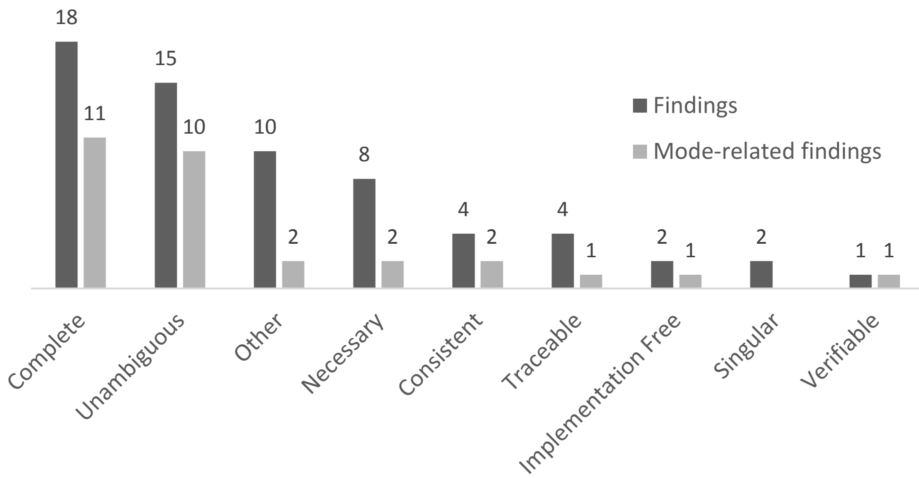

Fig. 2. Number of findings and mode-related findings per defect type.

RQ1: How many defects in NL requirements specifications mention modes? In summary, 30 of the 64 review findings that we inspected addressed issues related to a mode. That accounts for over 46% of all considered findings.
RQ2: Which types of requirements defects are issued by mode-related findings? As shown in Fig. 2, the largest number of mode-related findings reside in the defect types *complete* and *unambiguous*. In both classes, more than 60% of the findings are mode-related, which is the highest ratio of all defect types.[3] To illustrate typical findings of these defect types, we give two examples:

1. Defect type = unambiguous: *What does "active" in this case mean? That a warning is active or that the possibility to warn is active. Where is the definition of "active", "activation", etc.?*
2. Defect type = complete: *Missing requirement: During initialization of the vehicle, the optical icons must be activated.*

In the first example, the reviewer criticizes the ambiguous use of the mode `active`. She provides two possible interpretations of the term. In the second example, the reviewer points to a missing requirement that covers desired behavior in a special situation (mode) of the system, namely the `initialization` mode. The defect type *other*, which is the third most frequent defect type, summarizes all findings that did not fit in any of the defect type categories. Most of the findings in that category address the wording of a requirement.

[3] *Verifiable* is neglegted due to the small sample size.

RQ3: Which types of problems categorize the mode-related findings? Our open categorization process resulted in 6 categories, which characterize different problems that are issued in the mode-related findings. These categories are summarized in Table 1.

Table 1. Problem categories. In brackets, the number of findings in that category.

Category	**Problem description**
P1 (8)	A specific mode/situation is not considered in the function.
P2 (8)	A precise mode definition is missing.
P3 (5)	The reaction of a function to a mode switch is unclear.
P4 (4)	It is unclear how modes are logically connected in a function.
P5 (2)	The mode definition must change according to the context.
P6 (2)	The validity of a mode is doubted.

4 Discussion

We performed this study as a pre-study to assess the feasibility of the study design and explore whether the results are encouraging. Based on the reported results, we evaluate both intents positively. To strengthen and generalize the results, we plan to apply the study to a larger set of requirements specifications.

From the results gained so far, we infer that requirements specification in natural language suffer from a missing precise definition and specification of a set of modes. When specific terms used as modes are not precisely defined, a reviewer, and consequently also a developer, may not exactly know what is meant by that term (*ambiguity*). Furthermore, a precise definition of modes, and especially, the values a mode might have, can point to situations that a requirements engineer did not consider (*completeness*). The problem categories identified in RQ3 can be used as criteria to evaluate approaches that focus on explicitly documenting modes. For example, the mode modeling approach we presented in [6] supports problem categories P1, P2, and P6, but not P5.

5 Threats to Validity

A major threat to the internal validity of the results is that the classification scheme we used to classify the review findings may be blurry. A possible consequence could be that the results are not reliable and also subject to researcher bias. We tried to mitigate this threat by two measures. First, we used an established classification scheme from a standard (ISO 29148 [3]) for the classification of defect types and a definition of modes that we discussed and validated with developers in a previous study [6]. Second, we performed the classification independently by two researchers. We achieved an inter-rater agreement in terms of Kohen's Kappa of 0.68 (*substantial agreement* [4]) for the classification of the defect type and 0.42 (*moderate agreement* [4]) for the relation to a mode.

Although we selected the two inspected functions randomly and did not pick specific functions that appeared particularly defect-prone, or mode-related, our results may not be generalizable due to the small sample size. However, as a first indication, when we presented the results to practitioners involved in the vehicle functions, they stated that the presented results are plausible.

6 Conclusions

In this paper, we presented an investigation of defects documented in review protocols of natural language requirements specifications for automotive systems. This investigation revealed that over 46% of the considered review findings addressed issues related to states of a system or its surrounding, which we call modes. Moreover, our analysis revealed that mode-related findings are predominantly represented in the defect types *unambiguous* and *complete*.

If we can reproduce and confirm the results on a larger set of review findings, we take this as a motivation to promote the explicit modeling of modes as an important means to increase the quality of NL requirements specification. Our idea is to capture modes that are relevant for more than one feature in a *mode model* [6]. Such a model provides precise and validated definitions of modes and captures them in a structured way. A requirements engineer who uses such a model, for example as a checklist, to formulate requirements may produce requirements, which are less ambiguous (due to the fixed definition of modes). Additionally, an extensive list of modes may inspire a requirements engineer to imagine different situations in which specific requirements apply. Thus, the requirements engineer may produce requirements specifications that are more complete. The evaluation of this hypothesis is subject to future research.

Existing approaches for state-based modeling of single features (e.g., [2, 5]) emphasize the benefits of requirements specifications structured by modes and also its relation to feature interaction. We advance this idea and propose creating a common model of system modes that applies to every feature of a multifunctional system. In a recent study [6], we provided first evidence that it is possible to elicit such a model for realistic systems and that they are still manageable with respect to their size. In that study, the elicited mode model for an entire system consisted of 75 modes. In a future study, we plan to evaluate whether these elicited modes cover the modes that were issued in the review findings.

References

1. Broy, M.: Multifunctional software systems: Structured modeling and specification of functional requirements. Science of Computer Programming 75(12) (2010)
2. Dietrich, D., Atlee, J.M.: A mode-based pattern for feature requirements, and a generic feature interface. In: Proceedings of the 21st IEEE International Requirements Engineering Conference (RE'13) (2013)
3. ISO/IEC/IEEE: Systems and software engineering – Life cycle processes – Requirements engineering. ISO/IEC/IEEE 29148:2011(E), International Organization for Standardization, Geneva, Switzerland (2011)
4. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics 33(1) (1977)
5. Vogelsang, A.: An exploratory study on improving automotive function specifications. In: Proceedings of the 2nd International Workshop on Conducting Empirical Studies in Industry (CESI'14) (2014)
6. Vogelsang, A., Femmer, H., Winkler, C.: Systematic elicitation of mode models for multifunctional systems. In: Proceedings of the 23rd IEEE International Requirements Engineering Conference (RE'15) (2015)