

Martin Beckmann, Andreas Vogelsang, Christian Reuter

# A case study on a specification approach using activity diagrams in requirements documents

## Conference Object, Postprint

This version is available at <https://doi.org/10.14279/depositonce-6720>.



## Suggested Citation

Beckmann, Martin; Vogelsang, Andreas; Reuter, Christian (2017): A case study on a specification approach using activity diagrams in requirements documents. - In: Requirements Engineering Conference (RE), 2017 IEEE 25th International. - New York : IEEE. - pp. 253-262. - DOI: 10.1109/RE.2017.28. (*Postprint is cited, page numbers may differ.*)

## Terms of Use

© © 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# A Case Study on a Specification Approach using Activity Diagrams in Requirements Documents

Martin Beckmann  
Technische Universität Berlin, Germany  
martin.beckmann@tu-berlin.de

Andreas Vogelsang  
Technische Universität Berlin, Germany  
andreas.vogelsang@tu-berlin.de

Christian Reuter  
Daimler AG, Germany  
christian.c.reuter@daimler.com

**Abstract**—Rising complexity of systems has long been a major challenge in requirements engineering. This manifests in more extensive and harder to understand requirements documents. At the Daimler AG, an approach is applied that combines the use of activity diagrams with natural language specifications to specify system functions. The approach starts with an activity diagram that is created to get an early overview. The contained information is then transferred to a textual requirements document, where details are added and the behavior is refined. While the approach aims to reduce efforts needed to understand a system’s behavior, the application of the approach itself causes new challenges on its own. By examining existing specifications at Daimler, we identified nine categories of inconsistencies and deviations between activity diagrams and their textual representations. In a case study, we examined one system in detail to assess how often these occur. In a follow-up survey, we presented instances of the categories to different stakeholders of the system and let them assess the categories regarding their severity. Our analysis indicates that a coexistence of textual and graphical representations of models without proper tool support results in inconsistencies and deviations that may cause severe maintenance costs or even provoke faults in subsequent development steps.

## I. INTRODUCTION

Complex software systems, which e.g. can be found in distributed embedded systems in automotive electronics, require model-based and system-oriented development approaches [1]. Using graphical models for specification manages complexity and improves reusability and analytical capabilities [2], [3]. Although graphical models provide a suitable means to specify and understand dependencies and procedural behavior of a system, in industry they are usually accompanied by a textual representation. Previous work has shown the need for a continuous systems engineering environment, where referring or constitutive documents are essential to work on complex software systems [4]. Also the combined use of graphical diagrams and textual descriptions is considered beneficial for the requirements management process [5], [6]. In addition, for industrial applications, tool support and model exchange for graphical models is still not standardised and, as a result, manufacturer/supplier handover is still performed by textual documents. This is especially important, since these textual documents often serve as the basis for legal considerations between the contractors [6], [7]. Also, due to different backgrounds of the stakeholders, not everyone is capable of understanding the graphical models [8]. Thus, the

information contained in a model needs to be written in words to be appropriately reviewable [9].

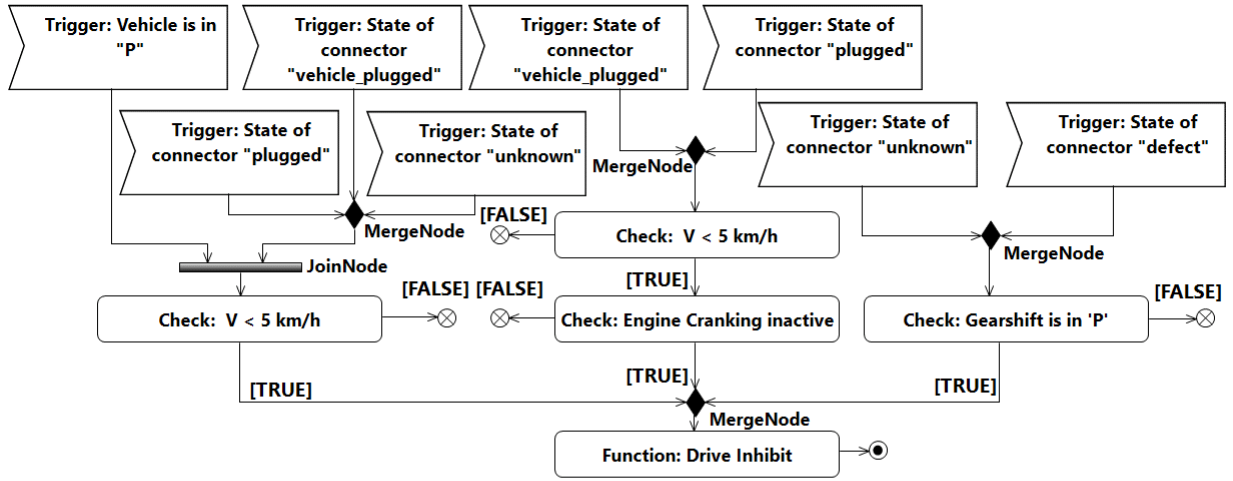
Daimler applies an approach, where, as a first step, a UML activity diagram [10] is created for each function to describe the function’s activation and deactivation by triggers and conditions. This kind of description is also known in literature to formulate textual natural language requirements [11]. Textual representations of the activity diagrams along with the diagrams themselves are then transferred into a requirements document for everyone to understand and for ongoing development. The transfer of the model into the requirements document is done manually. This is an error-prone task. Besides, the ongoing development using the requirements document might also cause inconsistencies between the activity diagrams and the document, in case the activity diagrams are not kept up-to-date.

We are interested in understanding what types of inconsistencies and quality issues are introduced by using activities and a textual specification alongside one another and how severe these issues are. If the approach itself introduces more severe issues than expected benefits, this is a strong argument for automated consistency checks and quality assurance.

For this purpose, we examined 36 vehicle functions of one system at Daimler that was specified by the introduced approach. As a result, a number of inconsistencies between the requirements document and the activity diagram were found.

All of these findings resulted in nine different categories of quality issues. We found occurrences of these categories in all of the examined functions. The categories are introduced in detail as well as the amount of findings in the examined system. Also we presented the quality issues to different stakeholders of the system, who assessed their severity. The occurrences, that are perceived as major quality issues, are present in 78% of the vehicle functions.

The paper’s structure is given in the following manner. The next section details the approach, that is used to specify the system’s functions. The third section introduces the nine categories, that were found examining the activity diagrams and their respective textual representation in a requirements document. In the fourth section the study design is explained. Section five presents the results of the study and the conducted survey. The sixth section discusses the results and possible means to avoid the discovered quality issues. The last section concludes this work.



(a) Activity diagram of the Function *Drive Inhibit*

ID	Text	Level	Type
1000	<b>1.1.1.1.1 Drive Inhibit</b>	6	Function
1236	State of connector "unknown" OR State of connector "defect" OR	7	Trigger
1237	Vehicle Gear selector is in position "P" AND	8	Check
1113	Engine Cranking inactive OR	8	Check
1111	State of connector "plugged on vehicle side" ("VEH_PLUGGED") OR "plugged on vehicle and EVSE side" ("PLUGGED"). OR	7	Trigger
1112	Vehicle velocity is below 5 km/h	8	Check
1114	Vehicle Gear selector is in position "P" OR	7	Trigger
1232	Vehicle velocity is below 5 km/h	8	Check
1233	State of connector "plugged on vehicle side" OR State of connector "plugged on vehicle and EVSE side" OR State of connector "unknown" AND	7	Trigger
1238	Vehicle velocity is below 5 km/h	8	Check

(b) Textual specification of the Function *Drive Inhibit*

Fig. 1: Activity diagram and the specification text of a function

## II. BACKGROUND

The Daimler approach used to specify functions of a system employs UML activity diagrams. These activity diagrams are the first step of specifying a new function. They are used to get an early overview of the desired function behavior with a special focus on the functions activation, execution conditions, functional paths, and deactivation. The information contained in the activity diagram as well as the activity diagram itself is then transferred to a textual requirements document. This transfer is necessary, since this textual requirements document is the central artefact for further development. Besides, the textual document contains additional and more detailed information as well as statements about its context, which relates this approach to Literate Modelling [8].

Fig. 1 shows an exemplar specification as we have found it at Daimler. The example consists of an activity diagram and

its textual representation in the requirements document. In the following, we will explain the example and also the contained quality issues. In the remainder of this work, an element refers to an entity contained in an activity diagram, whereas an object in the text refers to an entity contained in the requirements document.

Fig. 1a displays the activity diagram of the function *Drive Inhibit*. The actual behavior of the activated function is described in the *Action* node labeled with *Drive Inhibit* (bottom of the diagram). The function's activation is described by a combination of triggers and checks for conditions. For triggers, the *AcceptEventAction* element is used. The checks are modeled as *Action* elements. If the condition of a check is not fulfilled, the flow ends (*FlowFinal*). The triggers and checks are connected by *ControlNodes* such as *JoinNodes* and *MergeNodes*. *JoinNodes* act as synchronisation points and can

be interpreted as AND operators in terms of propositional logic. *MergeNodes* represent OR operators. Once the actual functionality of the function is executed, *ActivityFinal* elements designate the end of an activity.

The corresponding chapter in the textual requirements document is displayed in Fig. 1b. Each row in the document represents an object, which is described by a set of attributes (columns). The *ID* attribute contains a unique identifier of the object. The *Text* attribute is a textual description of the object and is supposed to be equal to the text of the corresponding element in the activity diagram. The *Level* is an attribute to structure the document hierarchically. It is derived from the structure of the activity diagram. The *Type* attribute of each text object is supposed to be equal to the type of its corresponding element in the diagram. These attributes are needed to display the relevant information of the activity diagram in the requirements document. Besides the given attributes, the document contains additional attributes used for further development.

There are many possibilities to display different aspects of an activity diagram as text. An exact textual representation as presented in [12] is not desirable, since it lacks proper readability and comprehensibility for those, unfamiliar with activity diagrams. Instead, the used textual representation focuses on the propositional logic, readability, and the recognition value of the structure of the activity diagram. This is implemented by copying the text of the elements of the diagram into distinct objects. Propositional logic operators such as OR and AND are used as strings in the *Text* attributes of the objects to realise the logic statements of the activity diagram. The operators at the end of an object's text connect the object with the following object on the same level of the document hierarchy. For instance, in Fig. 1b, the object with *ID 1236* is connected via an OR with the object with *ID 1111* because it is the next object on the same hierarchical level. Besides the propositional logic purposes, the different levels of the documents are used to display the belonging of the elements within the activity diagram. For example, the check *Vehicle Gear selector is in position "P"* (*ID 1237*) is executed after one of the triggers contained in the object with the *ID 1236* occurred. Hence, it appears one level below. This is important, since there might be more than one check associated with a set of triggers, as can be seen in the object in the text with *ID 1113*.

The transfer of information from the activity diagram to the requirements document is a manual process. This might lead to inconsistencies between the activity diagram and the requirements document and other quality issues as can be seen in Fig. 1. Amongst others, these inconsistencies and quality issues are presented in the next chapter.

### III. IDENTIFIED QUALITY ISSUES

A preliminary examination of a set of requirements documents at Daimler revealed a number of quality issues. The quality issues are inspired by standards such as ISO/IEC/IEEE 29148 [13], CMMI-Dev [14] and Automotive SPICE [15]. We grouped these quality issues into nine different categories. The

TABLE I: Categories of identified quality issues

Category name	Description
Missing Tracing	There is no information to trace an element to its corresponding object in the text or vice versa.
Missing Element/Object	Either the activity diagram or the requirements document contains entities, which the other does not contain.
Incorrect Logic	The propositional logic of the activity diagram deviates from the requirements document or the logic connections are not clear.
Textual Differences	Elements and their corresponding objects in the text exhibit textual differences.
Redundant Element	The activity diagram contains multiple elements, which have the same type and the same text.
Non Atomic Element/Object	Either an element or an object contains multiple statements. This might appear in both the requirements document or the activity diagram.
Wrong Placement	The placement of an element in the activity diagram does not match with the placement of the corresponding object in the requirements document.
Unnecessary Repetition	There are multiple objects in the requirements document, which are derived from one single element.
Wrong Type	The type of the element does not match the type of the corresponding object.

categories and their descriptions are listed in Table I. The categories cover the relation between the activity diagrams and the requirements document. Some of them only appear either in the diagram or the text, but still have an influence on the respective other artefact. We will explain some of the categories by examining the example in Fig. 1.

The category **Incorrect Logic** is present in the objects in the document with the *ID 1113* and *ID 1233*. Both objects end with an operator, for which it is not clear which object they refer to. Neither of them has a successor on the same level below their respective parent object. The object with the *ID 1236* is the parent object of two objects (*ID 1237*, *1113*) containing checks.

**Textual Differences** can be found (amongst others) between the triggers in the objects with the *ID 1111*, *1233* and their corresponding elements of the diagram.

There are multiple **Redundant Elements** in the diagram such as the checks *V < 5 km/h* and the triggers *State of connector "plugged"*. In this example, the appearance of redundant elements in the diagram can be avoided by inserting additional *ControlNodes* and restructuring the activity diagram [16].

While all elements of the diagram are atomic, the requirements document contains several **Non Atomic Objects** (*ID 1236*, *1111*, *1233*). These objects incorporate multiple assertions that are connected by propositional logical operators. This is both an issue in the requirements document as well as a

deviation between the activity diagram and the requirements document.

The elements in the diagram, corresponding to the object with the *ID 1236*, are followed by the diagram element *Check: Gearshift is in 'P'*. In the document the corresponding object (*ID 1236*) is the parent object of an additional check (*ID 1113*). The additional check in the document is elsewhere in the activity diagram. This situation is denoted as the category **Wrong Placement**.

The requirements document contains *Check:  $V < 5$  km/h* three times (*ID 1112, 1232, 1238*). But there are only two elements in the activity diagram. Hence, two of the objects in the document refer to one single element in the diagram. This is an instance of the **Unnecessary Repetition** category and can be avoided by grouping the objects accordingly.

We used these categories to find out how many quality issues occur in the vehicle functions of a system at Daimler and how much these occurrences influence the quality of the requirements document.

#### IV. STUDY DESIGN

To find out how often instances of the identified categories appear in a system and to understand how this system is impacted by these occurrences, we conducted a case study. We designed the study along the recommendations of Runeson and Höst [17]. Our research objective is:

**Research Objective:** We want to find out which problems the coexistence of textual and graphical representations of models implicate and how severe these problems are.

To reach this aim, we pursue four research questions (RQ):

**RQ1: How many occurrences of the categories can be found in a system?** To assess the influence of the occurrences of the categories, we need to know how many instances of each category occur. The number of occurrences of each category is one of the major contributing factors for the impact on the quality.

**RQ2: Are the stakeholders of the system aware of these occurrences?** We want to find out whether the stakeholders know about occurrences of existing quality issues. This gives us an idea on whether these occurrences have already been noticed. This is a first indication on how severe the occurrences are perceived.

**RQ3: Do the stakeholders agree that these occurrences are quality issues?** After we found out whether the stakeholders are aware of deviations and inconsistencies between activity diagrams and their textual representation, we want to know whether they agree with our assessment that a certain situation is in fact a quality issue. This is of interest since different backgrounds and responsibilities of the stakeholders might result in different opinions on what quality issues are.

**RQ4: How do the stakeholders assess the severity of these occurrences?** Besides the number of occurrences, the severity of an occurrence is the second major contributing factor of its impact on quality. Hence, the answer to this question is needed to evaluate the severity of each category.

**Study Object:** We examined a specific subsystem of a car developed at Daimler. The examined subsystem is responsible for charging the high-voltage batteries of Plug-in Hybrid Electric Vehicles and Battery Electric Vehicles. As such the system contains requirements that are relevant for safety as well as for usability. The system's requirements are documented in specification artefacts (activity diagrams and their textual representations) resulting from the approach described in Section II. The requirements document of the system contains a total number of 46 functions. In our study, we only considered 36 of these functions, since some functions were not specified using the approach and hence did not contain activity diagrams. Other functions were discarded because the corresponding text did not adhere to the pattern for the textual representation.

**Data Collection:** To answer RQ1, we manually searched for instances of the quality issue categories in all activity diagrams and their respective textual representation. To increase reliability and to avoid that occurrences are overseen, two researchers conducted this examination independently. The results were then compared and missing occurrences were complemented.

To answer RQ2 – RQ4, we conducted a survey amongst the stakeholders of the system that relates to the results of our manual document inspection. A total of seven stakeholders participated in the survey. Of the seven participants, three are authors of requirements documents for specific system components, two are responsible for testing, and one is an author of the requirements document of the system's functions. The last remaining participant is involved in developing the methodology that is used for the specification process.

As part of the survey, we presented two occurrences of each quality issue category as samples to the participants. The samples originated from specifications of several vehicle functions of the system. We selected actual samples of the system rather than abstract examples to improve the comprehensibility of each category and to give a better impression on the actual effect of the involved activity diagram and its corresponding textual representation. Each sample contains both of them. The issues in the activity diagram and the textual representation are highlighted by using colored frames. Besides, each sample is accompanied by a text explaining why the presented situation might have a negative effect on the quality. However, the concrete name of the category is not shown. This prevents the stakeholders from assessing the category rather than the concrete example. The rationale is to find out, whether the severity of different instances of one category is perceived differently. Most of the examined vehicle functions contain instances of multiple categories. Hence, some of the presented samples show the same vehicle function highlighting a different category each time. There was no specific order in which the samples were presented. However, the two samples of each category were never presented consecutively. The reason for this is to mitigate the influence of previous decisions.

The stakeholders were asked to answer the following survey questions (SQ) for each sample:

**SQ1: Were you aware of the existence of this finding?**

We first needed to know, whether the stakeholder had already recognised the presented situation of the sample. The participants could answer this by selecting *yes* or *no*. This aims at answering RQ2.

**SQ2: Do you think this sample is in fact a quality issue?**

This question is used to find out whether the stakeholder actually recognises the presented situation as a quality issue, now that it has been presented as such. The participants could answer this by selecting *yes* or *no*. The question aims at answering RQ3.

**SQ3: When would you fix this quality issue?** This was asked to assess the severity of the quality issue. We presented four options to answer this question to the stakeholders: *immediately* (during the same project iteration), *soon* (next time the function is edited), in the *long term* (when there is time to clean up the document), or *never*. The question aims at answering RQ4.

## V. STUDY RESULTS

### A. RQ1: Occurrences of Quality Issues

Table II shows the total number of occurrences we found for each category. The third column shows in how many functions we found quality issues of each category and the last column shows the average number of findings per function. The results show that we found at least 10 occurrences of each category in the 36 examined functions. Moreover, the Missing Tracing occurred in all elements of all functions, which means that we found no trace links to diagram elements at all. Secondly, we found missing elements or objects in the text or the diagram in 78% of the functions. In total, 126 elements and objects were missing, which accounts for 3.5 missing elements and objects per function on average. We found Incorrect Logic and Textual Differences in more than half of the examined functions. Textual Differences accounted for 43 findings in total. Wrong Type, Unnecessary Repetition, and Wrong Placement were the categories that appeared the least, although we still found them in about a quarter of all functions.

**Discussion:** The reported numbers show that a manual transition process between graphical activity diagrams and textual requirements documents bears a high risk of introducing deviations and inconsistencies, which we characterised as quality issues. Our analysis shows that this process is especially prone to missing out elements or objects, introducing incorrect logic, and textual differences. Whether these quality issues are really perceived as such by the stakeholders is examined in RQ2–RQ4.

### B. RQ2: Awareness of Quality Issues

The distribution of answers to SQ1 is displayed in Fig. 2. There are two bars for each category. Each bar represents the answers for one sample. In general, the presented samples were mostly unknown to the participants. There are five samples, where all participants mentioned that they were unaware of

TABLE II: Occurrences of quality issues per category ordered by frequency of occurrences in functions.

Category name	Findings	Number and ratio of functions with findings	Average number of findings per function
Missing Tracing	all <sup>1</sup>	36 (100 %)	-
Missing Element/Object	126	28 (78 %)	3.5
Incorrect Logic	29	22 (61 %)	0.8
Textual Differences	43	20 (56 %)	1.2
Redundant Element	24	15 (42 %)	0.66
Non Atomic Element/Object	18	14 (39 %)	0.5
Wrong Placement	18	10 (28 %)	0.5
Unnecessary Repetition	15	9 (25 %)	0.42
Wrong Type	10	8 (22 %)	0.28

their existence. For 12 samples, six out of seven participants stated that they were not aware of the existence. One sample belonging to the category Non Atomic Element/Object was known by two of the participants. It is worth noting that every time a situation was answered with *yes* at least once, a certain participant was always amongst those. This participant is the one involved in the development of the methodology.

**Discussion:** The answers to this question show that the stakeholders are mostly unaware of the presented occurrences. This fact explains, why we found these issues in the first place. Nevertheless we were quite surprised by these results. One possible explanation is that the selected participants were not involved in the development of functions from which we selected the samples. Since we selected the samples from a number of functions and a participant usually contributes to more than one function, this explanation is not very likely. An alternative explanation is that the selected samples belong to vehicle functions that are not frequently examined. Hence, their existence might have not been noticed. We had no information about the frequency of changes for functions. Another possibility is that the presented situations are not perceived as quality issues. Whether the samples are not perceived as quality issues so far or not at all is the subject of RQ3.

### C. RQ3: Agreement on Quality Issues

The answers to SQ2 are displayed in Fig. 3. The diagram is composed the same way as the diagram in Fig. 2. 14 out of the 18 samples were assessed to be quality issues by the majority of the participants. For three samples, all participants decided that these samples actually are quality issues. This applies to both samples of the category Wrong Placement. The other sample that all participants classified as a quality issue belongs to

<sup>1</sup>In the examined specifications, no tracing links between diagram elements and textual objects were defined.



Were you aware of the existence of this finding?

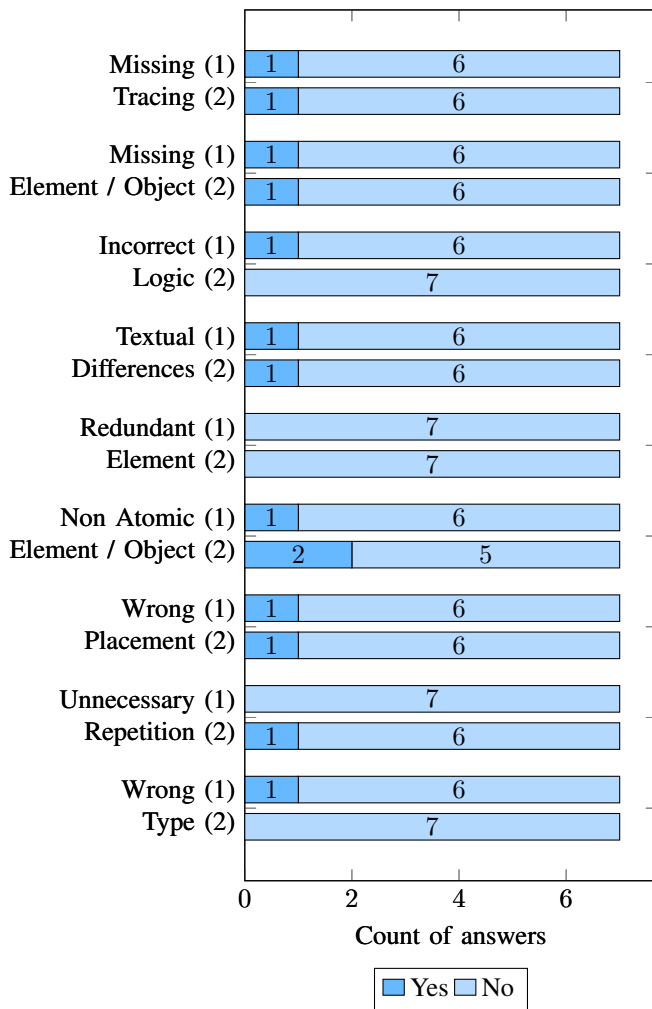


Fig. 2: Answers to SQ1

Do you think this sample is in fact a quality issue?

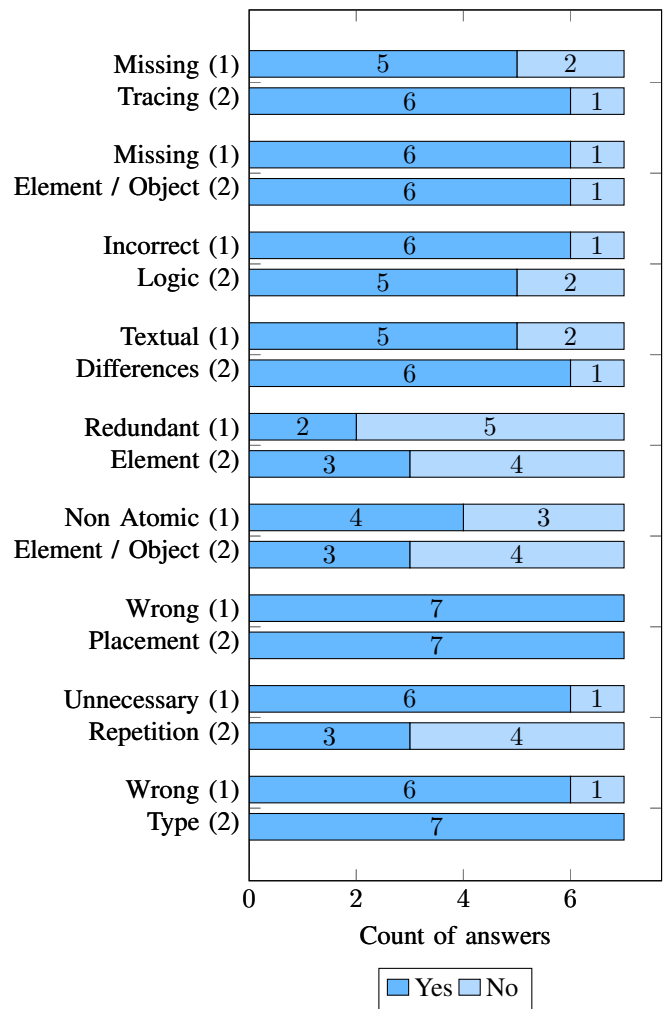


Fig. 3: Answers to SQ2

the category Wrong Type. Those two categories in addition to Missing Element/Object show the highest agreement amongst the participants to actually be quality issues. The samples with the lowest number of participants seeing them as quality issues are in the categories Redundant Element, Non Atomic Element/Object, and Unnecessary Repetition. Whereas most samples of one category were assessed similarly, the samples of category Unnecessary Repetition showed a large deviation. Its first sample is amongst those with the highest approval (six yes to one no), while the other is amongst the lowest (three yes to four no). We have no explanation for this result, since the two samples are very similar. Hence, further investigation is needed to assess, whether the stakeholders did not fully understand the presented situation or whether some of the stakeholders had a specific reason to assess the second sample differently.

**Discussion:** The answers to this question show that there is a high level of agreement that the samples of the categories Missing Element/Object, Wrong Placement, and Wrong Type in fact constitute quality issues. For the categories Redundant Element and Non Atomic Element/Object, many participants

assessed the identified samples as not being quality issues. This shows that the participants may have a different understanding of quality in these cases. Overall, there are only small differences between the samples within one category. This indicates that the perception might be the same for all other occurrences as well. How stakeholders assess the severity and whether the severity of the samples of one category are also similar is the subject of RQ4.

D. RQ4: Severity of Quality Issues

The answers to SQ3 are displayed in Fig. 4. As in the diagrams in Fig. 2 and Fig. 3 the answers for both samples of each category are displayed. The category, where the samples were perceived as most severe is Wrong Placement. At least five participants answered that they would fix these situations *immediately*. The remaining participants mentioned that they would fix them *soon*. For the categories Missing Element/Object, Textual Differences, and Wrong Placement no one answered with the option *never*. This means that all participants identified a need for improvement, which is in line with the

result of SQ2 where most participants assessed the samples of these categories as quality issues. The sample with the lowest severity is in the category Redundant Element (one *immediately*, two *soon*, one *long term*, three *never*). This is also in line with the results of SQ2. Some samples were assessed quite diverse. For example, the first sample of the category Non Atomic Element/Object would be fixed *immediately* by four participants, while two participants would *never* fix them. This sample consists of three propositional logic statements, that are all connected via an OR. In the requirements document all of the statements are contained in one single entry, while in the activity diagram, there are three distinct elements connected by a *MergeNode*. For the category Incorrect Logic, the severity of the two samples were assessed quite differently. While most participants agreed that they would fix the first sample *immediately*, two participants stated that they would *never* fix the second sample.

**Discussion:** The fact that, aside from one sample, the majority of the participants answered at least with *soon*, suggests that the identified categories are not only quality issues, but have to be considered for future development of the requirements document. However, some samples were rated with the option *never*. Especially for the samples of the categories Incorrect Logic it is interesting that some participants answered to *never* or only in the *long term* fix a quality issue, even for samples that reflect obvious deviations between the diagrams and the corresponding text (i.e., diagram and text describe different behavior). A possible explanation for this might be that these participants have a different understanding of the diagram's and text's semantics compared with us or that they just *use* them differently (e.g., they do not use the text to understand the function's behavior but only to look up some details). More than half of the participants answered to both samples of the categories Missing Element/Object and Wrong Placement with *immediately*. On top, no one answered with the option *never*. Therefore, we consider occurrences in the categories Missing Element/Object and Wrong Placement as major quality issues. Hence, 78% of the functions contain at least one major quality issue. This is also ratio of occurrences of the category Missing Element/Object. There was no occurrence of Wrong Placement without an occurrence of the category Missing Element/Object in the same function.

## VI. DISCUSSION

We conclude from our results that in this case a specification approach based on coexisting graphical and textual representation of specification models bears a high risk of introducing quality issues. More specifically, in our investigation of quality issues between activity diagrams and their textual representation, we assessed that Missing Tracing, Missing Element/Object, Textual Differences and Incorrect Logic are the most frequent quality issues.

Although most stakeholders were unaware of the occurrences that we presented to them, they agreed that those occurrences are in fact negatively impacting the quality of the requirements document. Since the samples of each category were always

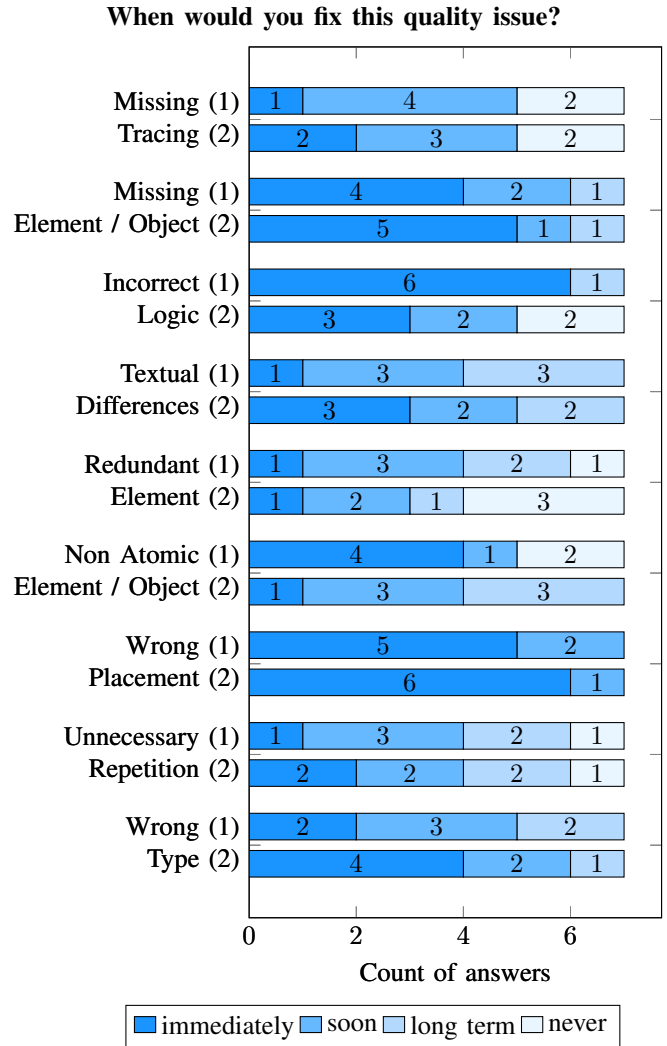


Fig. 4: Answers to SQ3

rated similarly by the participants, we may generalise this assessment to an assessment of the category itself. By this, we conclude that our defined categories Wrong Placement, Missing Element/Object, and Wrong Type definitively constitute quality issues. For the categories Redundant Element, Non Atomic Element/Object, and Unnecessary Repetition, the participant's opinions diverged. This challenges our initial hypothesis that findings of these categories are indeed quality issues.

The findings of RQ3 are consistent with the findings of RQ4. Categories Redundant Element and Non Atomic Element/Object are perceived as the least severe quality issues. In retrospect, this could also explain why participants were not aware of these quality issues since they did not perceive them as such so far. Also, the findings consistently suggest, that the categories Missing Element/Object and Wrong Placement are the most severe. This is especially important, since Missing Element/Object occurs the most often after Missing Tracing. Since Missing Tracing is a category appearing in every object of the document, it is hard to understand why it was rated as



a quality issue by so many. The findings of RQ4 also indicate, that there is a need for complete tracing between the artefacts of the two representations. The reason for the missing tracing is a consequence of the manual transition from the modeling tool to the RE management tool. This leads to the question why this approach was implemented without proper tool support in the first place. Possible solutions to this problem and the quality issues in general are presented in Section VI-A.

Initially, we also expected different perceptions of quality issues between participants of the different stakeholder groups. However, this was not confirmed in the study. The results were almost identical for participants with different stakeholder roles. There was only one exception. The expert on methodology was the only participant that was aware of most of the quality issues. At the same time that stakeholder only disagreed with us on three different samples originating from three different categories. Also that person only answered with *immediately* three times and is the person, who answered with *never* most often (four times).

Although it was not the scope of our research, comparing the timestamps of the activity diagrams with the dates of the baselines of the requirements document suggests that the activity diagrams are primarily used at the beginning. This might be caused by the specification process and is in accordance with other research findings [18].

#### A. Possible solutions

A possibility to reduce the number of quality issues, that arise by the manual transferal, might be the application of reviews as reported by Terzakis [19]. In order to avoid quality issues resulting from deviations between graphical and textual representations altogether, automatic approaches can be used to keep the diagram and the requirements document in sync. The generation of requirements documents from graphical models is an established approach [20]. Different approaches were suggested for specific graphical models. For instance, Maiden et al. [7] use *i\** models to derive requirements and De Landtsheer et al. [21] propose a similar approach for the KAOS goal-oriented method. Fockel and Holtmann [22] present a model-driven RE approach with tool support that provides synchronisation capabilities for the applied RE models and their textual representations. Still, these approaches are specialised to specific techniques during requirements elicitation and management.

Other than that, there are also approaches, that derive textual requirements or a structure for parts of requirements documents from different types of UML/SysML diagrams. Robinson-Mallett [23] shows how Statecharts and Block Diagrams can be used to create a structure for a requirements document. Berenbach [24] introduces an algorithm that derives a structure for a requirements document from use case diagrams. In an additional work [25], the possibility of synchronisation is mentioned, although it is limited to textual changes. In addition, the approach is restricted to diagrams that adhere to certain guidelines. Since all these approaches do not use activity diagrams, they are not applicable to the presented specification

approach. The approach presented by Drusinsky [26] supports activity diagrams, however, only for UML-1. Additionally, only the generation of actual requirements is addressed but not the creation of a requirements document structure.

Both the improvement of the manual transferal as well as automated approaches might benefit from an adjusted representation of the activities in the requirements document. A more sophisticated structure might mitigate some of the quality issues, while maintaining proper readability.

Another possibility is the use of Projectional Editors, which automatically edit different projections of a common underlying model, in this case the activity diagram and its textual representation. However, this possibility may require substantial efforts and experienced developers [27]. Hence, a custom-made and lightweight synchronisation solution might be well suited to prevent the mentioned quality issues for the used specification approach.

A more rigorous solution to the problem of inconsistent textual and graphical representations is to understand the development process as a stepwise refinement of natural language requirements to models that detail and formalise the original requirements [28]. In these processes, changes must be followed by a pipeline of updates along the chain of refinement models.

#### B. Threats to validity

The identification of quality issue candidates was performed manually. Manual processes are error-prone and also subjective to some degree. In order to mitigate this threat, two authors analysed the documents independently. To reduce the subjectivity, we developed a precise description of the quality issue candidates. We did not compute an inter-rater agreement, however, after the independent classification, we only had to discuss six occurrences for which the classification was different. Also, we cannot claim, that the quality issues we found cover all aspects of deviations between an activity and its textual representation.

We did not have any information about the development of the models and the documents over time. Including these information might lead to different classifications in some cases. For instance, certain findings that we classified as Missing Element/Object might actually be elements that were strongly altered over time so that we were not able to identify the relation between the elements any more. In this case, the finding would actually fall into the category Textual Differences.

Besides, the analysis was done without any explicit domain knowledge of the used system. This might have led to misinterpretations regarding the categories. This also affects the categories Missing Element/Object and Textual Differences as we might not have recognised mere textual changes as such. Instead these occurrences ended up in the category Missing Element/Object.

Due to the large number of findings, we only presented two representative findings of each category to the stakeholders. We used the assessment of these findings as proxy for an evaluation of the whole category. Our results show that the two samples

of each category, in general, were assessed similarly. Still, it is possible that the selected samples were perceived as more or less severe than other samples of the same category would have been. The participating stakeholders were selected by the third author who is also actively participating in the development of the examined system. We did not follow specific selection criteria, except that participants must work actively on the examined systems. However, the group of study participants only represent a subset of all people working actively with the requirements documents. Furthermore, not all of those we contacted, reported back to us in time. We originally contacted twelve participants of which seven answered our survey.

Our results indicate that quality issues arise from the presented specification approach that is used in some projects at Daimler. To answer our research questions, we only had access to one system developed with this approach. Hence, the generalisability of our findings are limited. Discussing the results with the stakeholders at least left the impression that the results are not surprising to them and that they would expect similar findings also in other systems developed with this approach.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we presented possible quality issues, that may arise when using a certain specification approach, that we encountered at Daimler. The approach incorporates UML activity diagrams in requirements documents. Those activity diagrams are accompanied by a textual representation of the diagrams. The textual representation is edited and further refined during ongoing development.

We conducted a case study on a real system. The purpose of this study was twofold. First we assessed the total number of occurrences of possible quality issues in the requirements document of the system. The second part is a survey amongst the stakeholders. The aim was to find out, whether they agree, that the quality issues we identified are in fact quality issues and how they rate the severity of preselected samples.

All of the examined functions were affected, since there was no tracing present between the activity diagram elements and the objects of the text. Other than that we found between 10 and 126 occurrences of each identified quality issue. The survey showed, that the stakeholders were unaware of the existing quality issues. Nevertheless, the majority of them agreed, that seven out of nine identified quality issues are in fact issues impacting the quality of the requirements document.

For all but one sample, the majority of the stakeholders saw the need to fix the quality issues at latest during the next time the function is edited. However, there are eight samples, where at least one stakeholder saw no need in fixing the issue.

Since there was only one system available, the generalisability is limited. The findings require additional validation by repeating the case study with a different system and more participants.

An aspect, that was out of scope of this work, is the influence of the identified quality issues on following development stages. The case study assessed the number of occurrences in a certain

requirements document and the severity of the quality issues, but not the resultant consequences. Hence, it needs to be addressed how these quality issues effect the development of the final product and future products, that reuse the existing requirements document.

## REFERENCES

- [1] M. Broy, "Challenges in automotive software engineering," in *Proceedings of the 28th international conference on Software engineering*. New York, NY, USA: ACM, 2006.
- [2] L. Apfelbaum and J. Doyle, "Model Based Testing," in *Software Quality Week Conference*, 1997.
- [3] A. Vogelsang, S. Eder, G. Hackenberg, M. Junker, and S. Teufl, "Supporting concurrent development of requirements and architecture: A model-based approach," in *Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD'14)*, 2014.
- [4] C. Reuter, "Variant Management as a Cross-Sectional Approach for a Continuous Systems Engineering Environment," in *Proceedings of the 8th Grazer Symposium Virtual Vehicle*, 2015.
- [5] A. M. Davis, *Just Enough Requirements Management: Where Software Development Meets Marketing*. New York, NY, USA: Dorset House Publishing Co., Inc., 2005.
- [6] E. Sikora, B. Tenbergen, and K. Pohl, "Industry needs and research directions in requirements engineering for embedded systems," *Requirements Engineering*, vol. 17, no. 1, 2012.
- [7] N. A. Maiden, S. Manning, S. Jones, and J. Greenwood, "Generating requirements from systems models using patterns: a case study," *Requirements Engineering*, vol. 10, no. 4, 2005.
- [8] J. Arlow, W. Emmerich, and J. Quinn, "Literate Modelling — Capturing Business Knowledge with the UML," in *International Conference on the Unified Modeling Language*. Springer, 1998.
- [9] R. F. Goldsmith, *Discovering Real Business Requirements for Software Project Success*. Artech House, 2004.
- [10] Object Management Group (OMG), "OMG Unified Modeling Language (OMG UML), Version 2.5," OMG Document Number formal/2015-03-01 (<http://www.omg.org/spec/UML/2.5/>), 2015.
- [11] D. Firesmith, "Generating complete, unambiguous, and verifiable requirements from stories, scenarios, and use cases," *Journal of Object Technology*, vol. 3, no. 10, 2004.
- [12] D. Flater, P. Martin, and M. Crane, "Rendering UML Activity Diagrams as Human-Readable Text," Tech. Rep., 2009.
- [13] The Institute of Electrical and Electronics Engineers, Inc., "ISO/IEC/IEEE 29148:2011, Systems and Software Engineering – Life cycle processes – Requirements Engineering," 2011.
- [14] CMMI Product Team, "CMMI for Development, Version 1.3," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., 2010. [Online]. Available: <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9661>
- [15] VDA QMC Working Group 13 / Automotive SIG, "Automotive SPICE Process Assessment / Reference Model," 2015.
- [16] M. Beckmann and A. Schlutter, "Automatische Duplikateliminierung in Aktivitätsdiagrammen von Fahrzeugfunktionen," in *INFORMATIK 2016, Lecture Notes in Informatics (LNI)*, 2016.
- [17] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, no. 2, 2009.
- [18] R. Hebig, T. Ho-Quang, G. Robles, M. Fernandez, and M. Chaudron, "The Quest for Open Source Projects that Use UML," in *19th International Conference on Model Driven Engineering Languages and Systems*, 2016.
- [19] J. Terzakis, "The Impact of Requirements on Software Quality across Three Product Generations," in *21st IEEE International Requirements Engineering Conference (RE)*. IEEE, 2013.
- [20] J. Nicolás and A. Toval, "On the generation of requirements specifications from software engineering models: A systematic literature review," *Information and Software Technology*, vol. 51, no. 9, 2009.
- [21] R. De Landtsheer, E. Letier, and A. Van Lamswerde, "Deriving tabular event-based specifications from goal-oriented requirements models," *Requirements Engineering*, vol. 9, no. 2, 2004.

- [22] M. Fockel and J. Holtmann, "A requirements engineering methodology combining models and controlled natural language," in *4th International Model-Driven Requirements Engineering Workshop (MoDRE)*. IEEE, 2014.
- [23] C. L. Robinson-Mallett, "An approach on integrating models and textual specifications," in *2nd International Model-Driven Requirements Engineering Workshop (MoDRE)*. IEEE, 2012.
- [24] B. Berenbach, "The Automated Extraction of Requirements from UML Models," in *11th IEEE International Requirements Engineering Conference (RE)*. IEEE, 2003.
- [25] B. Berenbach, "Comparison of UML and Text based Requirements Engineering," in *Companion to the 19th Annual ACM SIGPLAN Conference on Object-oriented Programming Systems, Languages, and Applications*, ser. OOPSLA '04. New York, NY, USA: ACM, 2004.
- [26] D. Drusinsky, "From UML activity diagrams to specification requirements," in *IEEE International Conference on System of Systems Engineering (SoSE)*, 2008.
- [27] T. Berger, M. Völter, H. P. Jensen, T. Dangprasert, and J. Siegmund, "Efficiency of projectional editing: A controlled experiment," in *24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE)*, 2016.
- [28] W. Böhm, M. Junker, A. Vogelsang, S. Teufel, R. Pinger, and K. Rahn, "A formal systems engineering approach in practice: An experience report," in *1st International Workshop on Software Engineering Research and Industrial Practices (SER&IPs'14)*, 2014.