INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.275

Automatic derivation of use case diagrams from interrelated natural language requirements

Simon Schleifer [1,✉], Adriana Lungu [2], Benjamin Kruse [2], Sebastiaan van Putten [2], Stefan Goetz [1] and Sandro Wartzack [1]

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany, [2] AUDI AG, Germany

✉ schleifer@mfk.fau.de

Abstract

Transferring natural language requirements to use case diagrams helps to avoid inherent ambiguities. However, this is usually a manual, time-consuming task that can be accelerated by utilizing Artificial Intelligence in terms of Natural Language Processing. Thus, this contribution proposes a conceptual framework for automatically grouping interrelated functional requirements and deriving use case diagrams by combining formerly isolated approaches. Moreover, the latter are evaluated by a qualitative potential analysis to support their future industrial application.

Keywords: requirements management, model-based systems engineering (MBSE), artificial intelligence (AI), natural language processing (NLP), use case diagrams

1. Introduction

In the ever-increasing complexity of technical products, the quantity of system requirements is inevitably rising. These requirements mark the starting point for the development process, resulting in a large amount of both functional requirements (FRs) and non-functional requirements (NFRs). In requirements engineering (RE), their documentation is typically done through the use of natural language (Pohl and Rupp, 2021). The main advantages are the comprehension without special training as well as the versatility of its application. Nevertheless, problems arise due to ambiguity and deficient controllability of complexity (Pohl and Rupp, 2021). To mitigate these limitations, model-based documentation is becoming increasingly popular (Pohl and Rupp, 2021). The use of Model-Based Systems Engineering (MBSE) is one potential implementation of this type of documentation. MBSE allows to handle increasing complexity and decreases ambiguities, thereby enhancing the product quality (Walden et al., 2015). Requirements can be formalized with the help of a modelling language such as the Systems Modeling Language (SysML). By utilizing requirement diagrams, it is possible to represent different kinds of relationships between requirements (Friedenthal et al., 2015). Nevertheless, this representation lacks a proper model-based definition since the natural language requirements remain unchanged within a requirement element (Salado and Wach, 2019). An alternative approach combines use case diagrams and activity diagrams to replace natural language requirements (Pohl and Rupp, 2021). The natural language requirements at the outset are only linked to the model-based description to maintain traceability.

This approach is applied within the industrial requirements engineering (Pohl and Rupp, 2021). First, use cases are derived from multiple functional requirements that define a system's functionality. Then, use case diagrams provide a black box view of the main functionality. These diagrams illustrate the entities associated with a specific functionality. For this purpose, actors, external systems, system

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press

boundaries, and the use case name are modelled with their relationships. In the industrial context, non-exclusive grouping of the available requirement data, especially of FRs, into functional units is necessary. Additionally, NFRs should be assigned to these groups serving as quality criteria. The activity diagram then describes the specific behaviour of a use case in detail, potentially representing a white box view (Pohl and Rupp, 2021). This practice can be time-consuming and could be accelerated through automation using Artificial Intelligence (AI). Therefore, methods of Natural Language Processing (NLP) are applied to enable automatic processing of natural language requirements. This includes traditional rule-based approaches and machine learning (Sonbol et al., 2022). Especially latter can make use of data originating from ongoing development. In the industrial context, this includes already grouped requirements and related use case diagrams as well as the assignment to superordinate system functionalities which comprise multiple use cases. Against the background of a model-based formalization of system requirements, the objective of this paper is a contribution towards the automatic derivation of use case diagrams from multiple interrelated requirements expressed in natural language. For this purpose, methods from NLP are investigated.

The remainder of this paper is structured as follows. In Section 2 the state of the art is represented. Section 3 illustrates the current research need. This is followed by the proposed framework to derive use case diagrams against the background of multiple related FRs in Section 4. The paper closes with a discussion in Section 5 and a conclusion in Section 6.

2. State of the art

NLP is a common technique within RE with ongoing research interest (Sonbol et al., 2022), leading to numerous previous related works. The following subsections present relevant contributions within requirement grouping (Section 2.1) and entity derivation (Section 2.2).

2.1. Grouping of natural language requirements

A grouping based on similar FRs is essential, to derive use cases described in the requirement data. Several publications study the categorization of natural language requirements. However, little research covers FRs in particular. Generally, it can be distinguished between classification and clustering. Classification on the one hand is a supervised learning method meaning that all training data is labelled with corresponding classes. A classifier is then able to assign unseen data to the predefined classes. On the other hand, clustering is an unsupervised learning method whose main goal is the identification of structures within unlabelled data. Therefore, a suitable distance measure is utilised (Ertel, 2017).

In terms of classification Sonbol et al. (2022) identify several research projects. These are mainly dealing with the distinction between FRs and NFRs, as the most commonly used classes within RE. Similar results are mentioned by Lopez-Hernandez et al. (2021). The authors also observe a widespread use of classification within NFRs. Further, Ott (2013) presents an automatic approach to classify requirements into manually defined topics to support review activities. For this, requirement data from previous projects is used as training data. Sangounpao and Muenchaisri (2019) combine an ontology with Naïve Bayes Classification to perform multi-class classification of FRs within the accounting domain. The potential classes are extracted from appropriate standards. The work of Chatterjee et al. (2020) deals with the identification of architectural significant FRs and their classification in different common classes, which are the result of a systematic literature review. Deep learning-based models are used to perform classification and, thus, time-consuming manual labelling of a training dataset is necessary. Lastly, Jp et al. (2022) developed a deep learning-based framework to classify FRs non-exclusively in predefined classes. For this purpose, fine-tuned word embeddings are combined with a convolutional neural network for classification.

Besides that, some research is done on the topic of clustering requirements. An early contribution to this by Niu and Easterbrook (2008) focuses on identifying overlapping clusters based on extracted linguistic features. The similarity of FRs is determined by co-occurring actors and actions, which are extracted via Part-Of-Speech (POS) tagging combined with custom rules. Afterwards, an overlapping partitioning cluster algorithm is applied. The following research projects focus on exclusive clustering. Casamayor et al. (2012) extract noun and verb phrases as possible actors and activities by utilizing POS tagging. Based on the extracted features, different clustering algorithms are compared. Salman et al. (2018) are

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press

using a vector space model to represent all words within the pre-processed FRs. Semantic similarity is measured by cosine similarity and a hierarchical clustering algorithm determines appropriate clusters. Kochbati et al. (2021) adopt this approach but use neural word embeddings to represent the vocabulary. Also, a more sophisticated similarity measure is applied before determining the clusters. Furthermore, Misra et al. (2016) aim to cluster requirements around identified themes. Latent semantic analysis is used to measure the semantic similarity of requirements and terms. The retrieved themes help the user to better understand the identified clusters. Finally, Gulle et al. (2020) try to assess topics within user stories. Therefore, they compare three different embedding and dimension reduction techniques. The best results are achieved by calculating Word Mover's Distance between the requirements. However, the evaluation of possible clusters is done manually.

2.2. Derivation of use case diagrams

Based on the functionally grouped requirements, an industry expert typically extracts all necessary elements to create the use case diagrams with regard to the natural language requirements. Several publications cover the topic of automating this process by utilizing NLP methods. These approaches most commonly originate in the software development domain. In general, two kinds of approaches can be distinguished.

On the one hand, rule-based methods utilize POS tagging to derive noun and verb phrases for actors and use case names, respectively. For example, Deeptimahanti and Sanyal (2011) use simple rules based on noun and verb phrases combined with a technique to extract prepositions from simple input sentences to derive associated elements of use case diagrams. Similarly, Elallaoui et al. (2018) utilize simple rules and POS tagging to extract use case diagrams from structured natural language. This approach is highly dependent on the correct use of a corresponding sentence template. The same applies to the rules based on noun and verb chunks according to Kochbati et al. (2021). The approach by Tiwari et al. (2019) is able to extract actors and use case names as well as preconditions, dependencies, basic and alternative flows, and post conditions based on a natural language FR expressed in multiple sentences. Yet, this approach is based on POS tagging and corresponding rules established by the authors. Additionally, a questionnaire is used to implement user feedback and enhance the overall quality. Recently, there have been efforts to combine traditional approaches based on POS tagging. Nasiri et al. (2021) add an ontology to better handle synonyms and to cover an “include” relationship. Further, Malik et al. (2023) take advantage of a network to represent discovered connections between the elements in order to create better use case diagrams. Within the scope of this paper, especially the publication by Park and Kim (2020) is relevant. The authors propose a rule-based approach to extract use case diagrams from multiple related FRs. For this, FRs are grouped according to their goal and subsequently analysed by an incremental sentence analysis. The individual tokens are identified and put in relation with the help of POS tagging and dependency parsing. This results in a shared model for each group of FRs. Furthermore, different relationships occurring in use case diagrams are identified. However, the suggested method lacks automation.

Other methods using Artificial Neural Network (ANN) exist for instance from Al-Hroob et al. (2018) who propose a first attempt to identify actors and use case. The semi-automatic approach uses the token, its POS tag and its dependency tag as input. Each token is further labelled by its role and used for a supervised training of an ANN with three hidden layers. Furthermore, approaches using machine learning have recently become more popular. Those are based on Named Entity Recognition (NER) and use manually labelled datasets to train custom NER models to detect actors and use case names (Tiwari et al., 2020). Vineetha and Samuel (2022) extend this approach to extract use case names consisting of multiple words. Except for Park and Kim (2020) all identified approaches have in common that one requirement leads to one set of elements for a single use case, meaning that the mapping between FRs and use cases is one-to-one.

3. Research need and methodology

Use case diagrams provide a model-based approach for describing system requirements and their functionality, by describing user interactions with a system. As shown in the previous section, there already exists research regarding the automatic creation of use case diagrams based on FRs utilizing

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press

methods from NLP. Nevertheless, the identified approaches are either related to isolated parts of this task, or are limited to requirement templates and embedded in software development such as the method proposed by Kochbati et al. (2021). Current research employs classification and clustering algorithms to group FRs. Methods to obtain overlapping groups are barely covered in existing research. Furthermore, a direct mapping between a single FR and a use case is prevalent. However, the industrial challenge is more complex. If FRs are grouped by classification solely, the ability to identify new use cases is neglected. Similarly, a pure clustering approach disregards existing knowledge in the ongoing development state. Further, a non-exclusive grouping is necessary to be beneficial for an industrial application. Additionally, a robust derivation of the necessary elements forming a use case diagram considering real industrial datasets needs to be investigated. This includes the use of existing knowledge such as glossaries. To the best knowledge of the authors, this is the first contribution to this topic considering the specific challenges of an industrial application within the automotive domain. The focus is on system requirements that are located in an early phase of the product development process and on a high level of abstraction with high heterogeneity. In summary, this leads to the following research question: How to automatically derive use case diagrams from multiple related natural language FRs with respect to the industry's needs?

To address this open question, this publication proposes a general AI-based framework to group and convert natural language FRs into SysML use case diagrams. For this purpose, methods of NLP are considered. The framework is based on the theoretical development of all required steps with the involvement of industry experts considering the aforementioned challenge. Related research is analysed to identify existing, yet isolated approaches. A qualitative potential analysis is conducted to assess the effective integration of existing industrial knowledge and data. Lastly, necessary possibilities for manual intervention are identified by interviewing experts from industry.

4. Framework for derivation of use case diagrams

To close the gap stated above, a new holistic framework for automatic derivation of use case diagrams from system requirements is proposed. The focus is placed on the combination of previously isolated approaches and their industrial applicability as well as on the non-exclusive grouping of FRs, which has not been addressed in prior studies. This includes the derivation of use case elements from a set of interrelated requirements rather than individual ones. The basis for subsequent behaviour modelling is provided by the novel framework utilizing NLP methods. The proposed framework shown in Figure 1 is described in detail in the following and provides the basis for the qualitative potential assessment of existing approaches in the light of the aforementioned industrial challenges. For better comprehensibility, the explanation is continuously illustrated by an academic example, see Figure 2.

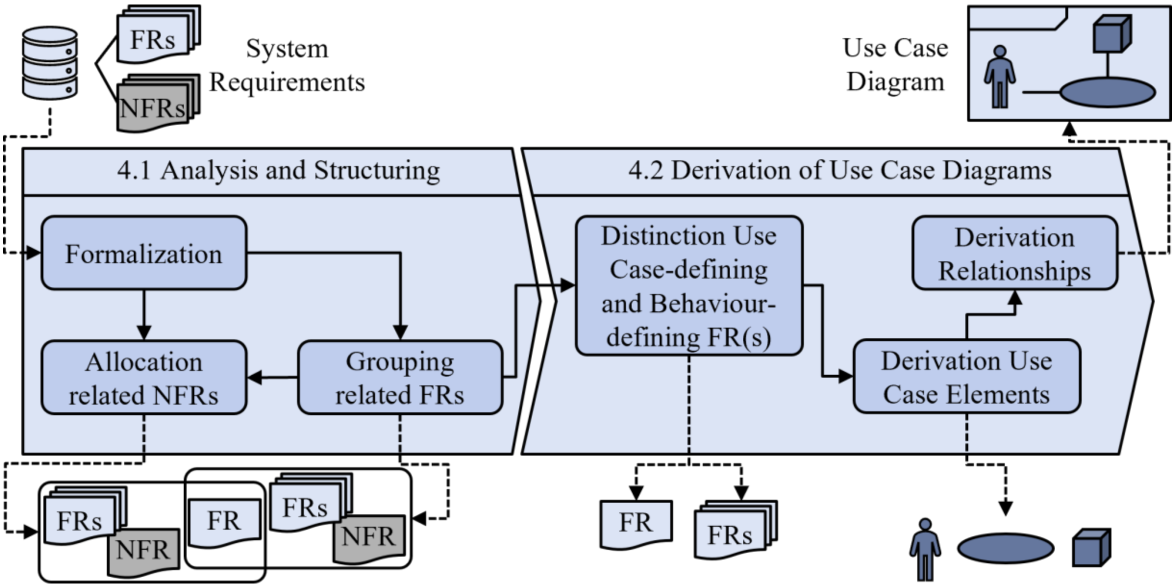

FRs
NFRs
System Requirements
Use Case Diagram
4.1 Analysis and Structuring
4.2 Derivation of Use Case Diagrams
Formalization
Allocation related NFRs
Grouping related FRs
Distinction Use Case-defining and Behaviour-defining FR(s)
Derivation Relationships
Derivation Use Case Elements
FRs
NFR
FR
FRs
NFR
FR
FRs

Figure 1. Framework to derive use case diagrams from interrelated natural language FRs

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press

First, the system requirement source containing functional relevant requirements is connected with the proposed framework. Next, the **formalization** of the natural language requirements is necessary to enable further processing. This can either be done by a rule-based approach or by using word embedding methods, respectively. Both approaches are well established within NLP. Based on the formalized requirements, functional groups within the FRs need to be identified (**grouping related FRs**). These potentially overlapping groups form the basis for the use cases as shown in Figure 2, where FR 2 is related to both use cases. Traditional AI methods such as classification and clustering need to be extended, to enable the usage of existing knowledge from the current development status and to reveal new use cases within the data. Additionally, non-exclusive grouping is necessary to cover FRs which address more than one use case. Because of the coarse articulation of system requirements, this overlap doesn't affect traceability negatively and is detailed in later steps like behaviour modelling. The **allocation of related NFRs** to the identified groups of FRs based on the requirement formalization is necessary to elaborate each use case in detail. The academic example in Figure 2 illustrates this by a performance requirement related to the use case "keyless access".

Following the preceding steps, each group is processed to derive the corresponding use case. Given that specific FRs primarily describe the behaviour of a use case, an identification of these is appropriate to support subsequent white box modelling. Hence, each identified functional group of requirements is **distinguished into use case-defining and behaviour-defining FRs**. This can be accomplished by rule-based approaches investigating the sentence structure or by using classification methods. Since all FRs potentially contribute to the upcoming identification of use case diagram elements, all requirements serve as input for the **derivation of use case elements**. This is achieved by utilizing either rules based on POS tagging and dependency parsing or custom NER models. Potential results from a NER model are highlighted for FR 1 in Figure 2, revealing use case name, actor, and (external) system. Using this result combined with the corresponding group of related requirements, all relationships need to be determined to complete the use case diagram (**derivation relationships**). To cover relationships between use cases, such as “include” and “extend” (cf. Figure 2), related use cases need to be combined. For this purpose, the known assignment of requirements to higher-level system functionalities within the training data can be used. Finally, the obtained black box representation of the use case diagrams offers insights into individual functionalities and interacting entities such as actors and external systems.

To guarantee effective integration into the development process, significant intervention options within the framework are determined through an interview with system architects. Three meaningful points of interaction emerge. First, a verification of the functional groups is necessary to enhance further processing. For this, a keyword-based summary of the group is beneficial. The framework should enable manual fine-tuning, too. Second, controlling the derived use case elements is required. The framework might detect new actors which need to be validated by the system architect or may result in changes of existing sets of actors. Lastly, the final use case diagram must be validated and possibly modified. Adding verified and modified results to the training data allows incremental improvement of the framework.

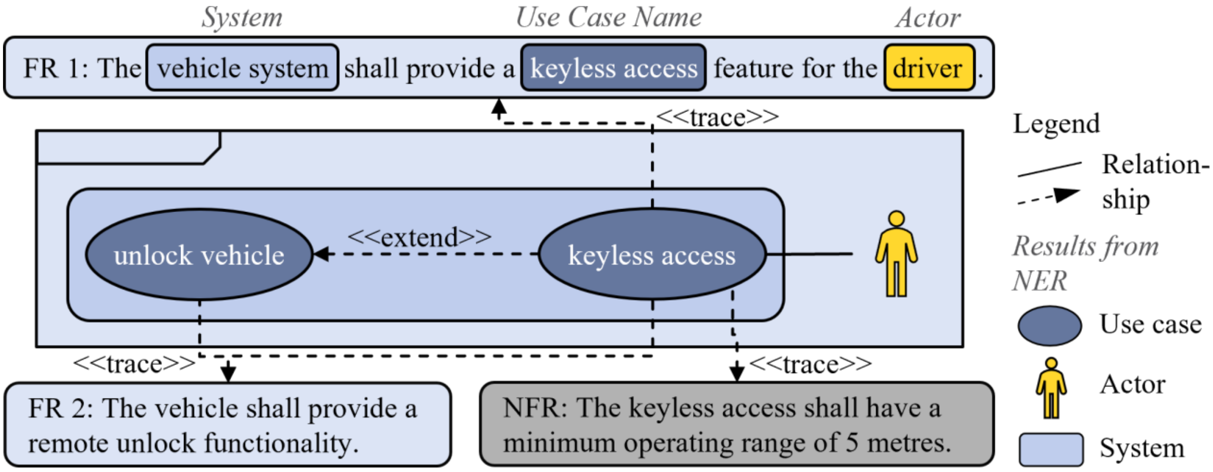

System
Use Case Name
Actor
FR 1: The vehicle system shall provide a keyless access feature for the driver.
<<trace>>
unlock vehicle
<<extend>>
keyless access
<<trace>>
<<trace>>
FR 2: The vehicle shall provide a remote unlock functionality.
NFR: The keyless access shall have a minimum operating range of 5 metres.
Legend
Relation-ship
Results from NER
Use case
Actor
System

Figure 2. Academic example illustrating the proposed framework

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press

The following subsections evaluate the utilization of pre-existing methods from literature in the light of the proposed framework.

4.1. Analysis and structuring

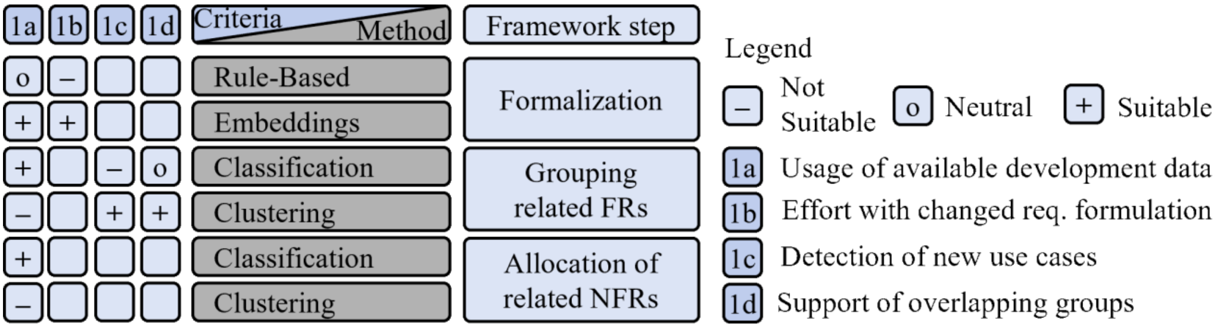
As already mentioned, a **formalization** method forms the basis for the grouping. It is therefore important to evaluate, to which extend the large amount of natural language requirements available in (automotive) engineering projects can be utilized. The identified approaches either use rule-based formalization or embedding techniques. Thus, these core concepts need to be evaluated considering the following assessment criteria: (1a) usage of available development data and (1b) effort with changed requirement formulation. On the one hand, rule-based approaches require a set of rules. In the industry, requirement templates and boilerplates are frequently used (Pohl and Rupp, 2021). Nevertheless, strict adherence to these rules is not always observed, and variations are explicitly accepted as long as the content is unambiguously conveyed. This potentially leads to a heterogeneous set of requirements (van Remmen et al., 2023). That implies that rule-based formalizations demand a significant number of rules, thus resulting in a considerable amount of manual labour initially and in case of changing requirement styles. Existing data is only used to establish the set of rules. Consequently, rule-based formalizations are perceived as time-consuming and therefore impractical. On the other hand, word embeddings can be obtained by using the totality of the development data as training input. It is worth mentioning, that Chatterjee et al. (2020) and Jp et al. (2022) report improved results in the subsequent grouping step with the implementation of self-trained embedding models. On the contrary, Gulle et al. (2020) achieve superior results with pre-trained embeddings compared to self-trained. This can be explained by the strong correlation between the quality of self-trained embeddings and the training set size. Since requirement sets in the automotive domain are large, the quality of embeddings is estimated to profit significantly by fine tuning pre-trained embedding models to get domain-specific models. Further, changes to the style of the requirement data can be covered by smaller changes to the pre-processing method. In summary, it can be stated that, given the context of industrial data volumes and variability, embedding methods are expected to have greater potential than rule-based formalization.

After formalizing, the FRs are categorised based on **functional groups** or use cases, respectively. In order to form suitable assessment criteria, the main objectives of the grouping stage comprise (1a), (1c) detection of new use cases, and (1d) support of overlapping functional groups. Existing research examines techniques relating to classification and clustering. As classification is part of supervised learning, all available labelled data is considered during training. Further, overlapping classification is well established in forms of multiclass classification approaches such as one-vs.-rest. Nevertheless, a major problem arises since classification is not able to perceive new groups. Hence, classification is not suitable for grouping of FRs. Clustering, conversely, automatically recognises groups within datasets, without being restricted to manually pre-determined categories. Some existing clustering techniques also support overlapping groups. However, clustering has the disadvantage of neglecting existing training data, as it is an unsupervised learning technique. Summing up, neither of the established grouping methods covers the specific demands of an industry application completely. Thus, the choice between the two methods should be tailored to the project, taking into account the available data and the significance of uncovering new groups.

Lastly, the **allocation of related NFRs** is assessed with regard to criterion (1a). The usage of classification offers the opportunity to include existing development data as labelled training sets. This step can be solved by classification since possible groups are defined in the previous step. An alternative is the usage of clustering. For this, a combination with the functional grouping step is necessary. Thus, existing development data can't be utilized during the allocation of NFRs either. In summary, the allocation of NFRs by classification has a higher potential than using clustering.

Figure 3 gives an overview of afore assessed core concepts concerning the first three framework steps.

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press



1a	1b	1c	1d	Criteria / Method	Framework step
o	–			Rule-Based	Formalization
+	+			Embeddings	
+		–	o	Classification	Grouping related FRs
–		+	+	Clustering	
+				Classification	Allocation of related NFRs
–				Clustering	

Legend
– Not Suitable o Neutral + Suitable
1a Usage of available development data
1b Effort with changed req. formulation
1c Detection of new use cases
1d Support of overlapping groups

Figure 3. Summary of the potential analysis for "Analysis and Structuring"

4.2. Derivation of use case diagrams

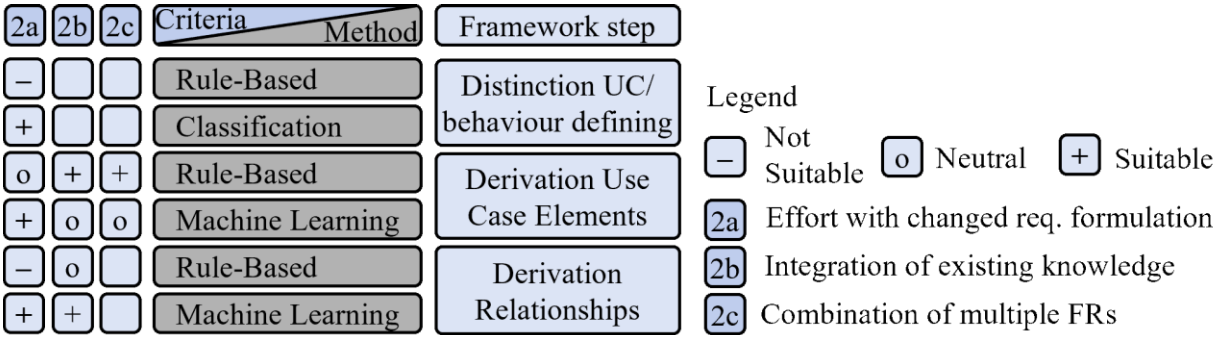
Based on the identified groups, a differentiation is established between **use case-defining and behaviour-defining FRs**, as illustrated in Figure 1. The primary assessment criterion is based on the utilization of existing development data (2a). For this task, a rule-based or machine-learning-based NLP approach can be chosen. As this activity is currently not documented in the development, it is not feasible to employ pre-labelled training data to train a classifier. Nevertheless, classification is estimated to perform better in this context. First, word embeddings can be reused from previous steps. Second, a classification approach is able to detect similarities between the two classes, that might not be apparent through manual rule formulation. However, manual effort exists by labelling the training data.

Following the proposed framework in Figure 1, the **derivation of use case diagram elements** forms the next step with a direct impact on the black box representation. The following criteria are included in the evaluation: (2a), (2b) integration of existing knowledge (e.g., glossaries), and (2c) combination of multiple FRs. The previously identified approaches comprise rule-based and machine learning methods. Similar to the previous section, rule-based approaches heavily rely on the adherence to requirement templates. Such approaches are well established to derive use case elements and, thus, a large variety of rule sets exist. Furthermore, some research aims to include existing knowledge by integrating an ontology. This is advantageous as ontologies often exist within the industrial context and may require only some preparation. It is also possible to combine FRs and the derived use case diagram elements by rule-based approaches. However, FRs with divergent formulations impede effective usage within large industry datasets. Machine learning based approaches comprise ANN and NER. With respect to assessment criterion (2a), it should be noted that these are supervised learning methods making labelled training data vital. The existing development state can't be used because a direct link between a single use case diagram element and a requirement is non-existing. To establish the required training dataset, manual labelling is necessary. However, labelling for large requirement sets is estimated to be less demanding than establishing rules for a rule-based approach. Since the integration of existing knowledge and the combination of multiple FRs in one use case diagram is based on previously identified elements, the methods proposed in rule-based approaches can be transferred to machine learning-based methods, as well. In conclusion, rule-based approaches are more common and, thus, offer solutions that exceed mere element derivation. However, as labelling of the existing development data manually is considered as less laborious than deriving a large rule set, machine learning methods are deemed to have greater potential.

Lastly, possible approaches to **derive relationships** in the context of use case diagrams are assessed. For this, assessment criteria (2a) and (2b) are considered, whereas the latter also includes existing use case diagrams. Rule-based approaches to determine relationships between elements are well established within the identified research. Changes to the requirement template or large variety within the formulations can require complex rules and increased effort to adapt to alteration. Some approaches also integrate existing knowledge by utilizing ontologies. Neither of the presented methods includes pre-existing modelling from ongoing development. Machine learning can contribute to mitigate this restriction. It is possible to use existing use case diagrams as labels for requirements based on the ongoing development status. A machine learning algorithm can be trained thereby and circumvent the necessity of developing a rule set. This approach is highly rated as it ensures effective use of existing

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press

models and knowledge. Furthermore, it is likely to be simpler to adjust to altered formulations because the amount of manual labour is minimized. Therefore, it is predicted that machine learning has a greater potential than rule-based methods within the scope of industry datasets.
Figure 4 summarizes the assessment of the steps required to derive the use case diagram.



2a	2b	2c	Criteria / Method	Framework step
–			Rule-Based	Distinction UC/ behaviour defining
+			Classification	
o	+	+	Rule-Based	Derivation Use Case Elements
+	o	o	Machine Learning	
–	o		Rule-Based	Derivation Relationships
+	+		Machine Learning	

Legend

– Not Suitable o Neutral + Suitable

2a Effort with changed req. formulation

2b Integration of existing knowledge

2c Combination of multiple FRs

Figure 4. Summary of the potential analysis for "Derivation of Use Case Diagrams"

5. Discussion

In this contribution a framework is presented, that outlines the necessary steps to derive use case diagrams from system requirements. The primary objective of the framework is to establish links between existing research efforts in the context of RE and MBSE and the specific needs of an industrial application in the engineering domain. An existing approach by Kochbati et al. (2021) is tailored to requirement templates in the software engineering context and neglects detecting overlapping functional groups. Both limitations are overcome by the here proposed framework.
Especially the overlapping clustering based on functional similarity is a key finding of this contribution as this meets the industrial challenge. Thus, the proposed framework is an extension to existing approaches, such as Park and Kim (2020), that do not consider an automatic derivation. The identification of use cases from system requirements is an important step in industrial RE activities and benefits from automation. Moreover, it comprises a basis for subsequent white box modelling executed to detail the use case's behaviour. It is beneficial to integrate the distinction between use case-defining and behaviour-defining requirements, even if both requirements potentially contribute to the use case diagram. A reason for this is the existing formalization of the requirements, which can be used as the basis for automatic grouping. Further, the overall understanding of a functional group can be enhanced, given that detailed insights in the use case behaviour are already provided.
The usability of the framework is ensured by discussions with industry experts. They indicate the need for the possibility for manual verification and adjustments to guarantee effective usage and enhance the overall system understanding. The potential analysis contributes to the research question by assessing prevalent NLP and AI methods in literature in the context of the specific industrial challenges. This reveals, that rule-based approaches are potentially restricted as requirement sets in the industry are not homogeneous and, thus, require large rule sets. In contrast, machine learning approaches require labelled training data. In most cases, the labelling is a manual task, since the data is usually not already annotated.
Limitations of this work concern the potential analysis. The evaluation is based on the qualitative assessment of the authors and has not yet been validated by implementation with real datasets from the industry. Especially the consideration between rule-based and machine learning-based approaches is highly dependent on the actual dataset and requires further validation for specific requirement sets. Finally, the entire framework needs to be investigated and validated in industrial applications to eliminate potential shortcomings.

6. Conclusion and outlook

The contribution emphasises the lack of a common approach to derive use case diagrams from interrelated natural language requirements in an engineering context. Thus, the proposed framework contributes to the automatic derivation thereof with respect to the automotive industry's needs. It

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press

incorporates non-exclusive grouping of FRs, the usage of available development information, points for potential interventions derived from interviews with industry experts. The framework merges existing isolated contributions to this topic, mainly utilizing rule-based and machine learning-based approaches. By assessing these approaches for each step with respect to demands occurring in an industrial context, the limitations of existing approaches are demonstrated.

The proposed framework reveals the need for further research. In this way, semi-supervised clustering (SSC) (Cai et al., 2023) could be utilized for grouping FRs to solve the conflict between reuse of existing knowledge and necessary flexibility. Further, active learning (AL) can be used to reduce the amount of labelled training data required and, thus, enable a more efficient application of machine learning methods. In particular, pool-based AL (Settles, 2009) can be utilized as a large amount of unlabelled data is available whereas only a small portion is labelled.

After all, the framework forms the basis for the upcoming behaviour modelling. Activity diagrams are used for this to detail and document the behaviour described in the requirements and grouped by use cases. This will be subject of future research.

Acknowledgement

The authors would like to thank AUDI AG for supporting and funding the research project.

References

Al-Hroob, A., Imam, A.T. and Al-Heisa, R. (2018), “The use of artificial neural networks for extracting actions and actors from requirements document”, *Information and Software Technology*, Vol. 101, pp. 1–15. https://doi.org/10.1016/j.infsof.2018.04.010

Cai, J., Hao, J., Yang, H., Zhao, X. and Yang, Y. (2023), “A review on semi-supervised clustering”, *Information Sciences*, Vol. 632, pp. 164–200. https://doi.org/10.1016/j.ins.2023.02.088

Casamayor, A., Godoy, D. and Campo, M. (2012), “Functional grouping of natural language requirements for assistance in architectural software design”, *Knowledge-Based Systems*, Vol. 30, pp. 78–86. https://doi.org/10.1016/j.knosys.2011.12.009

Chatterjee, R., Ahmed, A. and Anish, P.R. (2020), “Identification and Classification of Architecturally Significant Functional Requirements”, *IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), Zurich, Switzland, September 1, 2020*, IEEE, pp. 9–17. https://doi.org/10.1109/AIRE51212.2020.00008

Deeptimahanti, D.K. and Sanyal, R. (2011), “Semi-automatic generation of UML models from natural language requirements”, *ISEC'11: Proceedings of the 4th India Software Engineering Conference, Thiruvananthapuram Kerala, India, February 24-27, 2011*, Association for Computing Machinery, New York, NY, USA, pp. 165–174. https://doi.org/10.1145/1953355.1953378

Elallaoui, M., Nafil, K. and Touahni, R. (2018), “Automatic Transformation of User Stories into UML Use Case Diagrams using NLP Techniques”, *Procedia Computer Science*, Vol. 130, pp. 42–49. https://doi.org/10.1016/j.procs.2018.04.010

Ertel, W. (2017), *Introduction to Artificial Intelligence*, Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-58487-4

Friedenthal, S., Moore, A. and Steiner, R. (2015), *A Practical Guide to SysML: The Systems Modeling Language*, Elsevier MK, Amsterdam; Boston. https://doi.org/10.1016/c2013-0-14457-1

Gulle, K.J., Ford, N., Ebel, P., Brokhausen, F. and Vogelsang, A. (2020), “Topic Modeling on User Stories using Word Mover’s Distance”, *IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), Zurich, Switzland, September 1, 2020*, IEEE, pp. 52–60. https://doi.org/10.1109/AIRE51212.2020.00015

Jp, S., Menon, V.K., Soman, K. and Ojha, A.K.R. (2022), “A Non-Exclusive Multi-Class Convolutional Neural Network for the Classification of Functional Requirements in AUTOSAR Software Requirement Specification Text”, *IEEE Access*, Vol. 10, pp. 117707–117714. https://doi.org/10.1109/ACCESS.2022.3217752

Kochbati, T., Li, S., Gérard, S. and Mraidha, C. (2021), “From User Stories to Models: A Machine Learning Empowered Automation”, *Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development - MODELSWARD, February 8-10, 2021*, SciTePress, pp. 28–40. https://doi.org/10.5220/0010197800280040

Lopez-Hernandez, D.A., Octavio Ocharan-Hernandez, J., Mezura-Montes, E. and Sanchez-Garcia, A.J. (2021), “Automatic Classification of Software Requirements using Artificial Neural Networks: A Systematic

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press

Literature Review", *9th International Conference in Software Engineering Research and Innovation (CONISOFT), San Diega, CA, USA, October 25-29, 2021,* IEEE, pp. 152–160. https://doi.org/10.1109/CONISOFT52520.2021.00030

Malik, M.I., Sindhu, M.A. and Abbasi, R.A. (2023), "Extraction of use case diagram elements using natural language processing and network science", *PLoS ONE*, Vol. 18 No. 6. https://doi.org/10.1371/journal.pone.0287502

Misra, J., Sengupta, S. and Podder, S. (2016), "Topic cohesion preserving requirements clustering", *RAISE'16: Proceedings of the 5th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, Austin, Texas, USA, May 14-22, 2016*, Association for Computing Machinery, New York, NY, USA, pp. 22–28. https://doi.org/10.1145/2896995.2896998

Nasiri, S., Rhazali, Y., Lahmer, M. and Adadi, A. (2021), "From User Stories to UML Diagrams Driven by Ontological and Production Model", *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 12 No. 6, pp. 333–340. https://doi.org/10.14569/IJACSA.2021.0120637

Niu, N. and Easterbrook, S. (2008), "On-demand cluster analysis for product line functional requirements", *12th International Software Product Line Conference, Limerick, Ireland, September 8-12, 2008*, IEEE, pp. 87–96. https://doi.org/10.1109/SPLC.2008.11

Ott, D. (2013), "Automatic requirement categorization of large natural language specifications at Mercedes-Benz for review improvements", *Requirements Engineering: Foundation for Software Quality, Lecture Notes in Computer Science*, Vol. 7830, pp. 50–64. https://doi.org/10.1007/978-3-642-37422-7_4

Park, B.K. and Kim, Y.C. (2020), "Effort estimation approach through extracting use cases via informal requirement specifications", *Applied Sciences*, Vol. 10 No. 9, https://doi.org/10.3390/app10093044

Pohl, K. and Rupp, C. (2021), *Basiswissen Requirements Engineering: Aus- Und Weiterbildung Nach IREB-Standard Zum Certified Professional for Requirements Engineering Foundation Level*, dpunkt.verlag, Heidelberg.

Salado, A. and Wach, P. (2019), "Constructing True Model-Based Requirements in SysML", *Systems*, Vol. 7 No. 2: 19, https://doi.org/10.3390/systems7020019

Salman, H.E., Hammad, M., Seriai, A.-D. and Al-Sbou, A. (2018), "Semantic clustering of functional requirements using agglomerative hierarchical clustering", *Information*, Vol. 9 No. 9: 222. https://doi.org/10.3390/info9090222

Sangounpao, K. and Muenchaisri, P. (2019), "Ontology-Based Naive Bayes Short Text Classification Method for a Small Dataset", *20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Toyama, Japan, July 8-11, 2019*, pp. 53–58, https://doi.org/10.1109/SNPD.2019.8935711

Settles, B. (2009), Active Learning Literature Survey, *CS Technical Reports (TR1648)*, University of Wisconsin-Madison.

Sonbol, R., Rebdawi, G. and Ghneim, N. (2022), "The Use of NLP-Based Text Representation Techniques to Support Requirement Engineering Tasks: A Systematic Mapping Review", *IEEE Access*, Vol. 10, pp. 62811–62830. https://doi.org/10.1109/ACCESS.2022.3182372

Tiwari, S., Ameta, D. and Banerjee, A. (2019), "An Approach to Identify Use Case Scenarios from Textual Requirements Specification", *ISEC'19: Proceedings of the 12th Innovations on Software Engineering Conference (formerly known as India Software Engineering Conference), Pune, India, February 14-16, 2019*, Association for Computing Machinery, New York, pp. 1–11. https://doi.org/10.1145/3299771.3299774

Tiwari, S., Rathore, S.S., Sagar, S. and Mirani, Y. (2020), "Identifying Use Case Elements from Textual Specification: A Preliminary Study", *IEEE 28th International Requirements Engineering Conference (RE), Zurich, Switzerland, August 31 - September 04, 2020,* IEEE, pp. 410–411. https://doi.org/10.1109/RE48521.2020.00059

Van Remmen, J.S., Horber, D., Lungu, A., Chang, F., Van Putten, S., Goetz, S. and Wartzack, S. (2023), "NATURAL LANGUAGE PROCESSING IN REQUIREMENTS ENGINEERING AND ITS CHALLENGES FOR REQUIREMENTS MODELLING IN THE ENGINEERING DESIGN DOMAIN", *Proceedings of the Design Society, Vol. 3: ICED23*, pp. 2765–2774. https://doi.org/10.1017/pds.2023.277

Vineetha, V.K. and Samuel, P. (2022), "A Multinomial Naïve Bayes Classifier for identifying Actors and Use Cases from Software Requirement Specification documents", *2nd International Conference on Intelligent Technologies (CONIT), Hubli, India, June 24-26 2022,* IEEE https://doi.org/10.1109/CONIT55038.2022.9848290

Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D., Shortell, T.M. (2015), *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th edition., Wiley, Hoboken, New Jersey.

https://doi.org/10.1017/pds.2024.275 Published online by Cambridge University Press