NATURAL LANGUAGE PROCESSING IN REQUIREMENTS ENGINEERING AND ITS CHALLENGES FOR REQUIREMENTS MODELLING IN THE ENGINEERING DESIGN DOMAIN

van Remmen, Judith Sophie (1);
Horber, Dennis (1);
Lungu, Adriana (2);
Chang, Fan (2);
van Putten, Sebastiaan (2);
Goetz, Stefan (1);
Wartzack, Sandro (1)

1: Friedrich-Alexander-Universität Erlangen-Nürnberg;
2: AUDI AG

ABSTRACT

Requirements represent a central element in product development. The large number of requirements inevitably results in an increased susceptibility to errors, an expenditure of time and development costs. The associated problems motivate the application of Artificial Intelligence in the form of Natural Language Processing (NLP). In Requirements Engineering one main task is the classification of requirements which serves as the input in architectural models e.g. in SysML. In mechanical engineering there is still little overview regarding the interface between requirements classification and modelling. This paper provides an overview of the requirement classes and entities used in the literature and analyses their utilisation in modelling. Existing requirements classes usually do not offer the flexibility to be transferred to other domains. However, basic structures can be adopted from those classifications. This enables a clear assignment of existing classes to object classes in modelling. Resulting from the conducted literature study the observed predominant focus of research on the software industry requires an extension of the existing requirement classes and entities to enable further use and transfer to mechanical engineering.

Keywords: Requirements, Artificial intelligence, Natural Language Processing, Systems Engineering (SE), Literature review

Contact:
van Remmen, Judith Sophie
Friedrich-Alexander-Universität Erlangen-Nürnberg
Germany
van_remmen@mfk.fau.de

Cite this article: van Remmen, J. S., Horber, D., Lungu, A., Chang, F., van Putten, S., Goetz, S., Wartzack, S. (2023) 'Natural Language Processing in Requirements Engineering and Its Challenges for Requirements Modelling in the Engineering Design Domain', in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.277

1 INTRODUCTION

In the context of mechanical engineering, a contemporary trend towards Model-Based Systems Engineering (MBSE) can be observed. As an integral part of MBSE, the Systems Modeling Language (SysML) becomes more and more widespread for the modelling of systems in industry (Gausemeier *et al.*, 2013). SysML offers the possibility to create behaviour diagrams such as use case or activity diagrams. Use case diagrams in particular are used in practice to represent requirements for complex systems. Exemplary elements of those diagrams are actors or use cases. Requirements usually serve as the basis for the development of complex products. Therefore, with the modelling possibility of requirements in the respective requirement diagrams, SysML offers a great advantage compared to other languages such as the Unified Modeling Language (UML). (OMG 2019)

In general, requirements represent a central element in product development, as they consider the expectations and demands of all relevant stakeholders. Their satisfactory fulfilment forms an essential basis for the market success of a developed system. The large number of requirements inevitably results in an increased susceptibility to errors as well as an increased expenditure of time and development costs (Vlas and Robinson, 2011). Requirements are often semantically inconsistent, ambiguous, incomplete or inaccurate (Denger *et al.*, 2003). As the studies by Adam *et al.* (2013) and FIR e.V. (2021) show, the majority of companies use formulations in natural language. In most of the companies, these are available in semi-formalized, natural language. A lot of the organizations use text-based use cases and user stories, and in over half of the organizations surveyed, natural language is used without any structuring templates. The associated problems mentioned above and the increasing complexity of systems motivate computer support and automation in Requirements Engineering (RE). For this purpose, Artificial Intelligence (AI) in the form of Natural Language Processing (NLP) is applied to process natural language (FIR e.V., 2021). Studies show that companies focus process improvements, cost reductions, time savings, and increases in quality through the use of AI and NLP (Reder, 2021; FIR e.V., 2021). In this context, a series of NLP steps are usually applied to extract linguistic features and information from requirements texts in order to implement a targeted task such as requirements classification (Hey *et al.*, 2020; Horber *et al.*, 2020).

Classified requirements can serve as the input for the requirements modelling with languages such as SysML, e.g., within manually or automated modelling tasks. Accordingly, the elicitation of the *Requirement Classes* is elementary to maximize the quality in the modelling. Figure 1 shows the correlations between requirements and modelling. On the left side of the figure, the requirements enter a specific *Transfer Model* respectively a specific NLP representation, such as an ontology, as input. Using this *Transfer Model*, which often serves to structure or specify the requirements, the requirements in turn enter the modelling. All in all, it is thus the case that the requirements directly flow into the modelling. These must therefore be chosen in such a way that mapping between requirement and *Object Classes* is possible, regardless of the chosen *Transfer Models*.

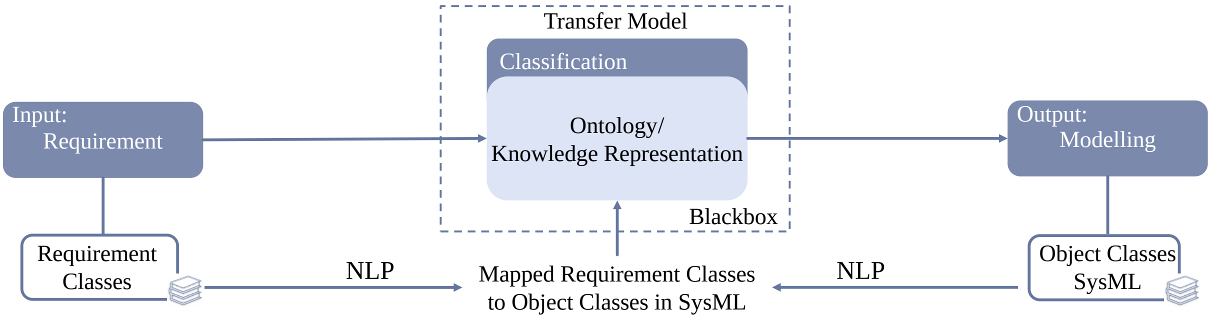

Figure 1. Illustration of the correlations between requirements and modelling

Current research deals with requirement classification, requirements modelling or *Knowledge Representation*, with less focus on the interdependencies between their methodical foundations. This contribution therefore identifies selected *Requirement Classes* and entities in the context of modelling in order to enable a possible reuse of existing classifications for mechanical engineering. The aim is to analyse the selected entities and class types, regardless of the technological implementation of the classification task, in order to be able to reuse them in other domains or application contexts. Hence, it is necessary to examine the relevant literature on requirements classification and modelling to identify

https://doi.org/10.1017/pds.2023.277 Published online by Cambridge University Press

these possible representations, gaps, and future research areas. Section 2 discusses related work in the area of systematic literature reviews on NLP in RE. Based on the derived research question and methodology in section 3, section 4 describes the specific methods used in the systematic literature review followed by a presentation and discussion of the results in section 5. Finally, the paper closes with a conclusion and an outlook in section 6.

2 RELATED WORK

Due to the different stakeholders or sources of requirements and the continuous concretisation throughout the product development process, different types of requirements are distinguished. Pohl and Rupp (2015) for instance differentiate between three types of requirements: functional, non-functional and constraints. In mechanical engineering, a formal structure or documentation, for instance via SysML, is usually preferred as a requirements classification. In addition, guidelines for writing the requirements are mostly used. In practice, however, even these directives cannot completely prevent unclear, redundant or ambiguous requirements (Krisch, 2017). Consequently, especially in software development, NLP methods are increasingly used to improve the systematic analysis, classification, and definition of requirements (FIR e.V., 2021). In this context, several literature studies investigating the use of NLP in different RE tasks are relevant to this paper, which are discussed in the following. In the current state of the art, literature studies are presented that provide a comprehensive overview of NLP in RE (Sonbol *et al.*, 2022; Zhao *et al.*, 2022), focus on a specific task (Raharjana *et al.*, 2021; Loniewski *et al.*, 2010) or describe specific NLP representations (Dermeval *et al.*, 2014).

The aim of the study by Sonbol *et al.* (2022) is to identify, categorize and analyse existing literature. In particular, it focuses on the use of syntactic and semantic aspects to represent software requirements. The papers were examined and mapped with respect to their NLP representations (e.g. lexical/syntactic, ontology, vector-based) and the respective addressed tasks (e.g. requirement analysis, requirements extraction, modelling). The main focus of the examined papers lies on the software area and the respective distinctions of the requirements into functional and non-functional and thereby on the specific implementation and the used representations and frameworks, e.g., described in Hey *et al.* (2020), Mir Khatian *et al.* (2021) and Rahman *et al.* (2019). Zhao *et al.* (2022) also provide a comprehensive review of the applications of NLP in RE research, focusing on the state of the literature and research and the NLP technologies used. Again, the primary focus is on software engineering. In addition to these studies, which provide a comprehensive overview of relevant literature, the study by Raharjana *et al.* (2021) focuses on the role of NLP in relation to user story specification and therefore one specific RE task (Requirements Elicitation). The study states that mostly NLP techniques were used to extract aspects of who, what, and why from user stories. Another study focusing on a specific RE problem is provided by Bozyiğit *et al.* (2021). The work presents a review of primary studies dealing with the automatic transformation of software requirements into conceptual models. It is stated that there is no feasible automated solution yet and that there is a need for more generic approaches including different diagram types in modelling. Another relevant study by Loniewski *et al.* (2010) provide an overview of RE techniques in the context of model-driven development, including cases where model transformation involves requirements expressed in natural language. In this context, Loniewski *et al.* (2010) note in their literature study that automated model transformations have been rarely used so far. However, the application of model-driven transformations at the requirements level could generate great benefits. An examination of ontologies and hence a specific NLP representation used in RE is provided by Dermeval *et al.* (2014) stating that ontologies support RE activities in academia as well as in industry.

The current state of the art shows, that previous work primarily focuses on software engineering. Implementation methods, tools or tasks are described for this specific domain, but do not focus the whole pipeline from requirements classification over *Knowledge Representation* to modelling. Additionally, it is often essential to specify *Requirement Classes* (e.g. Non-Functional Requirements (NFR)) and their characteristics (e.g. legal requirement) in order to be able to perform the modelling. Thereby, errors within modelling influence the requirements and vice versa (Alenazi *et al.*, 2019). The interdependencies between modelling and requirements can therefore not be neglected, but must be considered equally. This relation is necessary for the reuse of existing approaches from software engineering in the mechanical design domain.

https://doi.org/10.1017/pds.2023.277 Published online by Cambridge University Press

3 RESEARCH QUESTION AND METHOD

Previous literature studies, discussed in Section 2, mainly focus on the investigation of contributions that present specific NLP methods and representations. So far, classification and modelling tasks are usually considered separately from each other and cannot be detached, especially in mechanical engineering. The focus of this contribution is therefore to analyse the link between requirements and modelling as well as the entity and class types chosen so far (left and right side of Figure 1), independent of the *Knowledge Representations* (Figure 1 centre), which have been analysed in literature studies.

The structuring of requirements for documentation and modelling purposes is a key RE-activity in mechanical engineering (Pohl und Rupp 2015). Regardless of the *Transfer Models* (e.g. ontologies) chosen, it is therefore relevant to select the *Requirement Classes* in the classification task in such a way that modelling can be performed as efficiently as possible. If the requirements are already available in a structure that is suitable for modelling, for example through the use of structuring methods such as model-driven sentence templates, mapping the requirements to use cases in modelling is facilitated. Thus, the following research question (RQ), which forms the basis of this work, arise:

RQ: (1) How can requirements be classified using NLP (left side in Figure 1) and (2) what is the estimated reusability of these requirement classes in modelling (right side in Figure 1)?

Derived from the RQ the demand for an overview of contributions that map the relations between classification and modelling is addressed. By answering the RQ, the aim is to identify previous classification types in literature regardless of the specific *Knowledge Representation* (e.g. ontologies) and industry. Since requirements in early stages of development are specified independently of their implementation, the classes should not be limited to one domain (e.g. mechanical or software).

The methodical workflow used for the systematic literature review is shown in Figure 2. First, the understanding of the problem was sharpened within this framework in the step "clarification" and the search string was specified next. The systematic literature search was conducted according to the PRISMA guidelines (Page *et al.*, 2021). On the one hand, the results of the literature search should be the classes of requirements used in the literature, which are considered in the context of the modelling task. On the other hand, an estimation of the possible transfer of the chosen *Requirement Classes* to modelling and mechanical engineering is to be made. Thus, the research question will be answered, and the need for further research will then be derived. In order to answer the previously defined question, a systematic literature review is conducted.

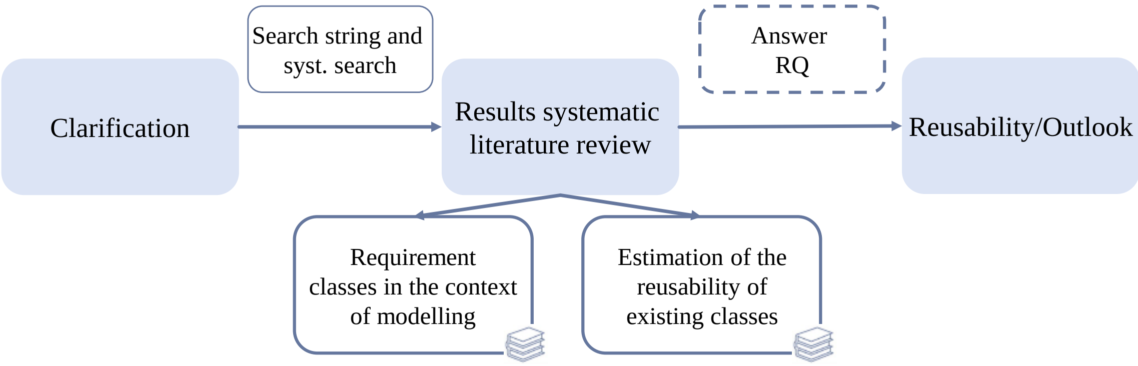

Figure 2. Illustration of the method of the systematic literature review

4 LITERATURE REVIEW

As mentioned before, the aim of the literature review is to identify the classification types which have been used in research so far in the context of the application of NLP in RE (left side of Figure 1). In particular, the review focuses on the identification of the chosen classes and therefore not on the chosen classification methods or *Transfer Models*. In total, up to October 2022 published literature was considered. Scopus is used as the primary source. The search includes scientific journal contributions, books and conference proceedings written in English. Within the scope of the clarification in Figure 2, the state of the art was analysed and the focus of the research was sharpened. Based on this, the search string for the systematic search is derived. The search string is shown in Figure 3 and represents the step after the clarification from Figure 2.

https://doi.org/10.1017/pds.2023.277 Published online by Cambridge University Press

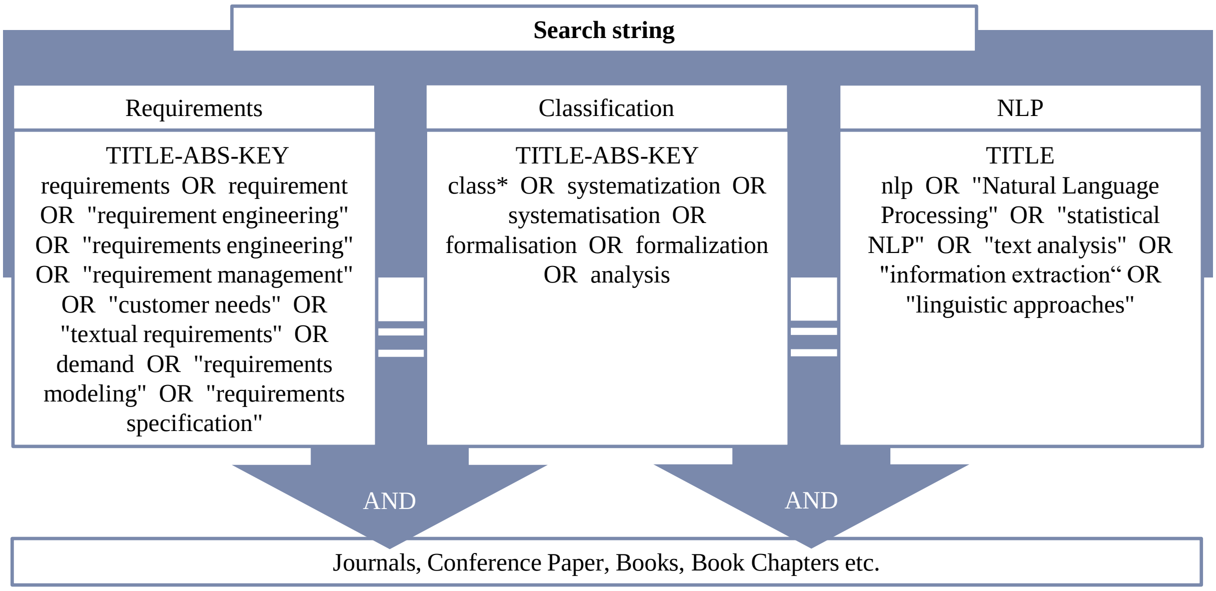

Figure 3. Search string within the systematic literature review

The search string focuses three main areas: Requirements, Classification and NLP. As the *Requirements Classes* are of particular relevance, the focus lays on classification tasks. As an additional constraint, classification using NLP is chosen as the frame condition.

The inclusion criterion consists of two aspects that require equal consideration: On the one hand, the requirements classification must necessarily be performed by NLP methods and, on the other hand, the RE task must be unambiguously attributable to classification, modelling, or an interface of the two. Accordingly, contributions that describe the articulation, elicitation, or tracing of requirements, e.g., through sentiment analysis, are not considered. The exclusion criterion is vice versa. Hence, contributions are excluded that either did not consider NLP methods, address a wrong RE task, and/or strongly focus on implementation or purely focus on building a *Knowledge Representation* based on existing classifications e.g. ontologies (Figure 1 centre). The selection process includes four steps:

1. Pre-selection 1: Analysis of the title, keywords and abstract | inclusion of the paper if the inclusion criterion 'NLP methods' is fulfilled.
2. Pre-selection 2: Re-reading of the keywords and abstract | inclusion if the criterion 'appropriate targeting and focusing of RE tasks' is met. Thus, addressing modelling or classification tasks.
3. Final selection: Reading of introduction and conclusion | include if both inclusion criteria are met
4. Reading of full paper | re-evaluation based on inclusion criteria.

5 RESULTS AND DISCUSSION

Figure 4 shows the number of papers in the individual steps of the literature study. It can be seen that initially 446 were detected using the search string described in section 4. After removing the duplicates, 444 titles were analysed according to their title and by reading the abstract and included or excluded according to the selection criteria described in section 4. Afterwards, the full papers were read and re-evaluated according to the established inclusion criteria. Thus, the systematic literature review identified 26 articles. The systematic literature search is subject to limitations. The results refer to the research conducted with the specified search terms and criteria (section 4). Therefore, the findings are limited by that and cannot be generalised. Nevertheless, the paper contributes to an extended overview of the state of the art on this specific topic and derives future research topics. In addition, an initial literature search was conducted to intensively analyse the related reviews mentioned in section 2 and their references, and to discuss them within the state of the art. An overview of the total of 26 contributions can be found in the appendix of this contribution. In order to identify the selected *Requirement Classes*, the contributions are analysed and the targeted RE tasks and *Object* and *Requirement Classes* are listed. The highlighted rows in the table added to the appendix are contributions that methodologically and thematically meet the inclusion criteria, but address the computer-assisted analysis of languages other than English, such as Arabic. In the following, the results of the research question are presented and analysed.

https://doi.org/10.1017/pds.2023.277 Published online by Cambridge University Press

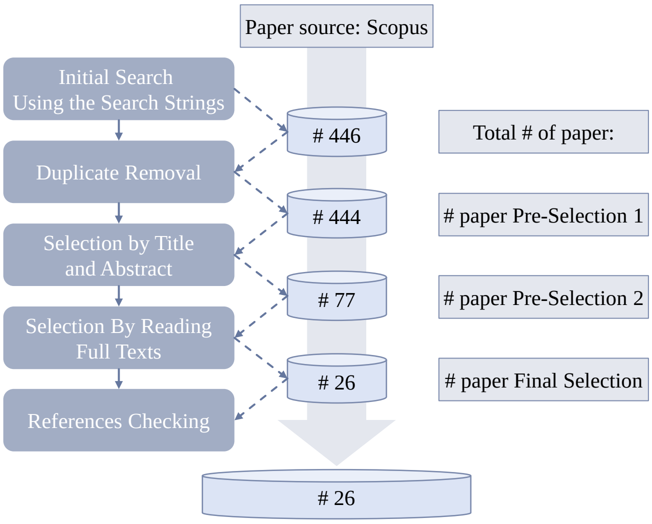

Figure 4. Selection of contributions during the systematic literature search

The resulting 26 papers are used to answer the research question. While 11 of these papers primarily address a classification task, 15 papers focus on NLP in the context of modelling. Moreover, 21 of the 26 papers can be assigned to the software engineering, which indicates that the research in the context of classification and modelling tasks primarily focuses on this domain (see section 2). Table 1 shows the identified requirement classes of exemplary contributions that focussed classification tasks using NLP (RQ-1). A more detailed overview is provided in the appendix of this contribution. Regarding the classification tasks, it is noticeable that these papers often perform a classification between FR and NFR (see also section 2). Furthermore, the focus is often on the further subclassification of NFRs into their quality aspects. Thereby, the estimated reusability/transferability addresses the extent of adaption needed for the further use in modelling. For the listed classes in Table 1, it is low (RQ-2), because the classification in FRs and NFRs is reasonable in the first step, but in terms of modelling, more fine-grained entities (sub-classification) would have to be defined. Thus, they need to be congruent with the *Object Classes* in the respective modelling language.

Table 1. Exemplary contributions and their respective classes of requirements as well as an assessment of their transferability to the mechanical engineering domain

Classes	Author	Transferability/ Re-Usability
FRs/NFRs and their quality aspects	Shehadeh *et al.*, 2021; Goyker et al., 2008; Shreda et al., 2021	Low
Actions and conditions	Anwar *et al.* 2020	Low
Security requirements (security levels such as confidentiality, integrity, availability, accountability, operational, access control)	Varenov and Gabdrahmanov 2021	Low
Complexity (high, medium, low)	Sundararajan *et al.* 2018	Low

As mentioned regarding the improvement of the transferability through sub-classes, e.g. Gröpler *et al.* (2021) distinguish between syntactic and semantic entities (action, action constraints, subject/object and signal, attributes, actor/component), which are also applied in SysML. Koscinski *et al.* (2021) and Shehadeh *et al.* (2021) also differentiate between entities such as subject, relation, object, descriptor, case and negation. Here the re-usability/transferability is high (RQ-2) due to the direct transferability to *Object Classes* in SysML. The majority of the papers focus on the software domain and they largely refer to the distinction between functional and non-functional requirements. In general, the specification of the classes according to syntactic and semantic criteria in different entities increases the potential for further use and simplify the transfer of the classes to the *Object Classes* in modelling.

With regard to modelling, a strong focus on the modelling language UML can be observed, which coincides with the focus on the software domain. SysML offers some advantages in contrast to UML due to the possibility of requirements modelling with designated requirement elements. However, it can be stated that the identified papers mostly focus exclusively on the classification or the modelling task

https://doi.org/10.1017/pds.2023.277 Published online by Cambridge University Press

respectively the automated model generation in UML (Drave *et al.*, 2020; OMG, 2019). Yet, it is necessary to consider the further use of the requirements already during the classification process. Therefore, a conceptual approach is necessary in the area of mechanical engineering in order to be able to convert the *Requirement Classes* into modelling objects as efficiently as possible. Apart from the focus on UML, the majority of the analysed contributions addresses the information extraction for the transformation of the requirements into use case and activity diagrams (A. Abdelnabi *et al.*, 2021; M. Maatuk and A. Abdelnabi, 2021). Overall, the hypothesized facts from the state of the art are confirmed with a strong focus in software engineering and thus the choice of classification into Functional Requirements (FRs) and NFRs. However, as can be seen, apart from the widespread classification into FRs and NFRs, there is a high demand for domain-specific language and classification models (e.g. mechanical). Overall, semantics is often assigned only a subordinate role, but it is often essential for domain-specific representation, as already shown in Sonbol *et al.* (2022). Yet, as described in the state of the art, unstructured requirements are often present that have not been written or documented according to any standards. So, more flexible approaches are necessary that also take further use into account when defining the entities or requirement types. The proposed literature review is limited since it did not include the identification of any papers that are specialised on a mapping method. In the future, there will be a need to focus specifically on and analyse existing mapping as well as interface concepts in order to derive a specific method that enables the transfer of *Requirements Classes* to *Object Classes* in mechanical engineering. However, the provided overview of classes used in the literature in Table 1 and the appendix answers RQ-1 and enables further analysis with regard to their use in modelling.

In terms of modelling, different object elements are provided depending on the diagram type. In the case of use case diagrams, the individual use cases are described textually by specifications. They represent the employment of the FRs as well as NFRs and can be specified by activity diagrams. In the automotive area, for example, requirements can furthermore be deduced from the activity diagrams, as shown by Anwar *et al.* (2020). To derive potential use cases from requirements, heuristics and rules can be applied that parse requirements into nouns, noun phrases, verbs, verb phrases, etc. (Alami *et al.*, 2017; Jaiwai and Sammapun, 2017; Shweta and Sanyal, 2020). Requirements can also be scanned for action verbs to derive use cases (Rago *et al.*, 2009). Actors, use cases, respectively components in SysML are linked via symbols to represent associations, generalizations and relationships. An activity diagram is used to describe the flow of a use case, yet it is suitable for modelling any activity within a system. An activity diagram specifies a series of activities. The detailed rules for how sentence blocks flow in an activity form the basis for the interpretation of an activity diagram. Thus, specific activities within a use case are modelled. Overall, it can be seen that the system requirements are the input for the derived use cases, and these in turn are the basis for the specification of activities. Consequently, a pure distinction between FRs and NFRs is often not sufficient in practice and therefore it is only possible to reuse existing *Requirement Classes* to a limited extent. There the answer to RQ-2 is that more finely granulated entities must be defined, which can then in turn be assigned as clearly as possible to elements in the chosen modelling language (Table 1). Here, a concept adapted from the state of the art or a new methodical approach is required for mapping the entities or requirement types to the object elements in the respective modelling language. For example, the use case "the user operates the cross-slide of the lathe in manual mode with little effort" could be mapped to several functions and especially to functions that are not explicitly mentioned in the use case, e.g. a good guidance of the cross-slide to prevent jamming.

Overall, it can be seen that a choice of entities based on the linguistic designations for word types would be possible in order to then enable a transfer or mapping to object elements in modelling via heuristics and rules. In summary, however, it can be stated that a pure distinction between FRs and NFRs needs to be further specified and reaches its limits, especially outside of software engineering.

6 CONCLUSION AND OUTLOOK

The results of the systematic literature review show that even when addressing mere classification or modelling tasks in RE, research in the field of software engineering dominates. Furthermore, it can be seen that in the literature, the left and right sides of Figure 1 are usually considered and examined separately from each other. However, it can be shown that the classification of requirements (left side Figure 1) and modelling (right side Figure 1) cannot be separated. This contribution focuses on the investigation of existing *Requirement Classes* from the literature and the examination of their further use in modelling. It could be shown that existing *Requirements Classes* and entities are only suitable for further use to a limited extent and that there is therefore a need for investigations that enable the mapping between requirements and *Object Classes* in modelling. All in all, a differentiation into NFRs and FRs is prevalent. Sporadically, word types or specific entities are assigned within the requirements depending on the use case. However, the current state of the art lacks a deeper analysis of semantics for domain-specific representation and more flexible approaches, especially with respect to unstructured or multilingual textual requirements. With focus on modelling, approaches for the automated generation of UML diagrams as well as use case and activity diagrams prevail. The answer to the research question shows that the mostly chosen requirement classifications of FRs and NFRs cannot be transferred without adapting or refining them. However, generic classifications in entities offer the possibility of a mapping to the object elements in SysML. The contribution provides guidance by listing existing classes of requirements and discussing them in terms of their further use. Based on this, the need for the development and analysis of concepts for transferability is derived.

Overall, it must further be analysed how domain-specific classifications, e.g. from software engineering, can be applied within other domains like the mechanical engineering domain. Since the domains interfere with each other, a common base in terms of a generic model is needed, especially for early stages of the RE process, where implementation-independent specifications are focused. Further classification into domain-specific subclasses may be feasible, but has to be investigated in future. Consequently, a generalised concept is required to map requirements to object elements in modelling languages such as SysML. Models for such a concept can possibly be adapted from software engineering, but have to be tailored to the requirement classification and its process in mechanical engineering. This prevents, that methods from the software industry are transferred to the mechanical engineering domain without adaption, which would lead to an unclear assignment of many requirements or requirement sentence parts to their respective (sub-) classes (e.g. legal requirements). Otherwise, application-specific isolated solutions are created. This in turn results in a lack of potential exploitation with regard to the reusability of methods and models.

ACKNOWLEDGMENTS

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

REFERENCES

A. Abdelnabi, E., M. Maatuk, A. and M. Abdelaziz, T. (2021), “An Algorithmic Approach for Generating Behavioral UML Models Using Natural Language Processing”. *ICEMIS'21*, Almaty Kazakhstan, 11 10 2021 13 10 2021, ACM, New York, NY, USA, pp. 1–6. http://dx.doi.org/10.1145/3492547.3492612.

Adam, S., Wünch, C. and Koch, M. (2013), *Ergebnisbericht „RE-Kompass 2013“*, IESE-Report Nr. 053.13/D, Fraunhofer IESE, Kaiserslautern.

Alami, N., Arman, N. and Khamyseh, F. (2017), “A semi-automated approach for generating sequence diagrams from Arabic user requirements using a natural language processing tool”. 2017 *ICIT*, Amman, Jordan, 17.05.2017 - 18.05.2017, IEEE, pp. 309–314. http://dx.doi.org/10.1109/ICITECH.2017.8080018.

Alenazi, M., Niu, N. and Savolainen, J. (2019), “SysML Modeling Mistakes and Their Impacts on Requirements”. 2019 *IEEE REW*, Jeju Island, Korea (South), 23.09.2019 - 27.09.2019, IEEE, pp. 14–23. http://dx.doi.org/10.1109/REW.2019.00010.

Anwar, M.W., Ahsan, I., Azam, F., Butt, W.H. and Rashid, M. (2020), “A Natural Language Processing (NLP) Framework for Embedded Systems to Automatically Extract Verification Aspects from Textual Design Requirements”. *ICCAE* 2020, Sydney NSW Australia, 14 02 2020 16 02 2020, ACM, New York, NY, USA, pp. 7–12. http://dx.doi.org/10.1145/3384613.3384619.

Bozyiğit, F., Aktaş, Ö. and Kılınç, D. (2021), “Linking software requirements and conceptual models: A systematic literature review”, *Engineering Science and Technology, an International Journal*, Vol. 24 No. 1, pp. 71–82. http://dx.doi.org/10.1016/j.jestch.2020.11.006.

Denger, C., Berry, D.M. and Kamsties, E. (2003), “Higher quality requirements specifications through natural language patterns”. *SwSTE'03*, Herzlia, Israel, 04.11.2003 - 05.11.2003, IEEE, pp. 80–90. http://dx.doi.org/10.1109/SWSTE.2003.1245428.

Dermeval, D., Vilela, J., Bittencourt, I.I., Castro, J., Isotani, S. and Brito, P. (2014), “A Systematic Review on the Use of Ontologies in Requirements Engineering”. 2014 *SBES*, Maceio, Brazil, 28.09.2014 - 03.10.2014, IEEE, pp. 1–10. http://dx.doi.org/10.1109/SBES.2014.13.

Drave, I., Rumpe, B., Wortmann, A., Berroth, J., Hoepfner, G., Jacobs, G., Spuetz, K., Zerwas, T., Guist, C. and Kohl, J. (2020), “Modeling mechanical functional architectures in SysML”. *MODELS '20 ACM/IEEE*,

Virtual Event Canada, 16 10 2020 23 10 2020, ACM, New York, NY, USA, pp. 79–89. http://dx.doi.org/10.1145/3365438.3410938.
FIR e.V. (2021), *Einsatz von Künstlicher Intelligenz zur Sprachverarbeitung: Anwendung von Natural-Language-Processing in der Industrie*, Bundesministerium für Wirtschaft und Energie (BMWi), Berlin.
Gausemeier, J., Dumitrescu, R., Steffen, D., Czaja, A., Wiederkehr, O. and Tschirner, C. (2013), *Studie Systems Engineering in der industriellen Praxis*, Heinz-Nixdorf-Institut, Fraunhofer IPT, Unity AG, Paderborn.
Gröpler, R., Sudhi, V., García, E. and Bergmann, A. (2021), "NLP-based requirements formalization for automatic test case generation". *CEUR Workshop Proceedings*, Berlin, pp. 18-30.
Hey, T., Keim, J., Koziolek, A. and Tichy, W.F. (2020), "NoRBERT: Transfer Learning for Requirements Classification". 2020 *IEEE 28th International Requirements Engineering Conference (RE)*, Zurich, Switzerland, 31.08.2020 - 04.09.2020, IEEE, pp. 169–179. http://dx.doi.org/10.1109/RE48521.2020.00028.
Horber, D., Schleich, B. and Wartzack, S. (2020), "CONCEPTUAL MODEL FOR (SEMI-) AUTOMATED DERIVATION OF EVALUATION CRITERIA IN REQUIREMENTS MODELLING", *Proceedings of the Design Society: DESIGN Conference*, Vol. 1, pp. 937–946. http://dx.doi.org/10.1017/dsd.2020.52.
Jaiwai, M. and Sammapun, U. (2017), "Extracting UML class diagrams from software requirements in Thai using NLP". 2017 *JCSSE*, NakhonSiThammarat, Thailand, 12.07.2017 - 14.07.2017, IEEE, pp. 1–5. http://dx.doi.org/10.1109/JCSSE.2017.8025938.
Koscinski, V., Gambardella, C., Gerstner, E., Zappavigna, M., Cassetti, J. and Mirakhorli, M. (2021), "A Natural Language Processing Technique for Formalization of Systems Requirement Specifications". 2021 *REW*, Notre Dame, IN, USA, 20.09.2021 - 24.09.2021, IEEE, pp. 350–356. http://dx.doi.org/10.1109/REW53955.2021.00062.
Krisch, J. (2017), "Sprachliche Kontrolle von Anforderungsdokumenten", Dissertation Universität Hildesheim, Hildesheim, 2017.
Loniewski, G., Insfran, E. and Abrahão, S. (2010), "A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development", In Petriu, D.C., Rouquette, N. and Haugen, Ø. (Eds.), *Model Driven Engineering Languages and Systems*, *Lecture Notes in Computer Science*, Vol. 6395, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 213–227. http://dx.doi.org/10.1007/978-3-642-16129-2_16.
M. Maatuk, A. and A. Abdelnabi, E. (2021), "Generating UML Use Case and Activity Diagrams Using NLP Techniques and Heuristics Rules". *DATA'21* 2021, Ma'an Jordan, 05 04 2021 07 04 2021, ACM, New York, NY, USA, pp. 271–277. http://dx.doi.org/10.1145/3460620.3460768.
Mir Khatian, V., Ali Arain, Q., Alenezi, M., Owais Raza, M., Shaikh, F. and Farah, I. (2021), "Comparative Analysis for Predicting Non-Functional Requirements using Supervised Machine Learning". 2021 *CAIDA*, Riyadh, Saudi Arabia, 06.04.2021 - 07.04.2021, IEEE, pp. 7–12. http://dx.doi.org/10.1109/CAIDA51941.2021.9425236.
OMG (2019), "Systems Modeling Language, SysML, v1.6".
Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P. and Moher, D. (2021), "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews", *BMJ (Clinical research ed.)*, Vol. 372, n71. http://dx.doi.org/10.1136/bmj.n71.
Pohl, K. and Rupp, C. (2015), *Requirements engineering fundamentals: A study guide for the certified professional for requirements engineering exam foundation level - IREB compliant, Rocky Nook computing*, 2nd edition, Rocky Nook, Santa Barbara, CA.
Rago, A., Abait, E., Marcos, C. and Diaz-Pace, A. (2009), "Early aspect identification from use cases using NLP and WSD techniques". *the 15th workshop*, Charlottesville, Virginia, USA, 03.03.2009 - 03.03.2009, ACM Press, New York, New York, USA, p. 19. http://dx.doi.org/10.1145/1509825.1509830.
Raharjana, I.K., Siahaan, D. and Fatichah, C. (2021), "User Stories and Natural Language Processing: A Systematic Literature Review", *IEEE Access*, Vol. 9, pp. 53811–53826. http://dx.doi.org/10.1109/ACCESS.2021.3070606.
Rahman, M.A., Haque, M.A., Tawhid, M.N.A. and Siddik, M.S. (2019), "Classifying non-functional requirements using RNN variants for quality software development". *the 3rd ACM SIGSOFT International Workshop*, Tallinn, Estonia, 27.08.2019 - 27.08.2019, ACM Press, New York, New York, USA, pp. 25–30. http://dx.doi.org/10.1145/3340482.3342745.
Reder, B. (2021), *Studie Machine Learning* 2021, IDG Business Media GmbH, München.
Shehadeh, K., Arman, N. and Khamayseh, F. (2021), "Semi-Automated Classification of Arabic User Requirements into Functional and Non-Functional Requirements using NLP Tools". 2021 *ICIT*, Amman, Jordan, 14.07.2021 - 15.07.2021, IEEE, pp. 527–532. http://dx.doi.org/10.1109/ICIT52682.2021.9491698.
Shweta and Sanyal, R. (2020), "Impact of passive and negative sentences in automatic generation of static UML diagram using NLP", *Journal of Intelligent & Fuzzy Systems*, Vol. 39 No. 2, pp. 2047–2059. http://dx.doi.org/10.3233/JIFS-179871.

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. http://dx.doi.org/10.1109/ACCESS.2022.3182372.

Sundararajan, M., Srikrishnan, P. and Nayak, K. (2018), "Requirements Complexity Definition and Classification using Natural Language Processing". 2018 *IC3I*, Gurgaon, India, 10.10.2018 - 12.10.2018, IEEE, pp. 76–80. http://dx.doi.org/10.1109/IC3I44769.2018.9007287.

Varenov, V. and Gabdrahmanov, A. (2021), "Security Requirements Classification into Groups Using NLP Transformers". 2021 *REW*, Notre Dame, IN, USA, 20.09.2021 - 24.09.2021, IEEE, pp. 444–450. http://dx.doi.org/10.1109/REW53955.2021.9714713.

Vlas, R. and Robinson, W.N. (2011), "A Rule-Based Natural Language Technique for Requirements Discovery and Classification in Open-Source Software Development Projects". 2011 *HICSS* 2011, Kauai, HI, 04.01.2011 - 07.01.2011, IEEE, pp. 1–10. http://dx.doi.org/10.1109/HICSS.2011.28.

Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K.J., Ajagbe, M.A., Chioasca, E.-V. and Batista-Navarro, R.T. (2022), "Natural Language Processing for Requirements Engineering", *ACM Computing Surveys*, Vol. 54 No. 3, pp. 1–41. http://dx.doi.org/10.1145/3444689.

APPENDIX

#	Author	Year	Title	Task	Diagram Type/Classes
1	Bashir, N. et al.	2021	Modeling Class Diagram using NLP in Object-Oriented Designing	MDL	Generation of class diagrams
2	Aydemir, F.B. et al.	2017	Towards aligning multi-concern models via NLP	MDL	Analysis of requirements diagrams
3	Almazroi, A.A. et al.	2021	Class Diagram Generation from Text Requirements: An Application of Natural Language Processing	MDL	Derivation of class diagrams
4	Rago, A. et al.	2009	Early aspect identification from use cases using NLP and WSD techniques	MDL	Identification of Use Cases
5	Jura, J. et al.	2022	Using NLP to analyze requirements for Agriculture 4.0 applications	MDL	Generation of Use Case and UML Class Diagrams
6	Abdelnabi, E.A. et al.	2021	An Algorithmic Approach for Generating Behavioral UML Models Using Natural Language Processing	MDL	Method for generating class, use case, activity, sequence, collaboration diagrams.
7	Beckmann, M. et al.	2018	Information extraction from high-level activity diagrams to support development tasks	MDL	Information extraction from UML activity diagrams to derive requirements and test cases
8	Shweta et al.	2020	Impact of passive and negative sentences in automatic generation of static UML diagram using NLP	MDL	Extraction of UML diagrams from unstructured FRs (especially in terms of multi-word expressions)
9	Ali, B.S. et al.	2011	An approach for crosscutting concern identification at requirements level using NLP	MDL	Distinction in FR and NFR and correlations (use cases)
10	Maatuk, A.M. et al.	2021	Generating UML use case and activity diagrams using NLP techniques and heuristics rules	MDL	Generation of Use Case and activity diagrams
11	Ibrahim, M. et al.	2010	Class diagram extraction from textual requirements using natural language processing (NLP) techniques	MDL	Generation of class diagrams (Use Cases, Activities)
12	Alkhader, Y. et al.	2006	Experimenting with extracting software requirements using NLP approach	MDL	Generation of class diagrams
13	Nassar, I.N. et al.	2015	Constructing activity diagrams from Arabic user requirements using Natural Language Processing tool	MDL	Rules for deriving activities for use in activity diagrams
14	Alami, N. et al.	2017	A semi-automated approach for generating sequence diagrams from Arabic user requirements using a natural language processing tool	MDL	Parsing of Requirements into Nouns, Noun Phrases, Verbs, Verb Phrases, etc. to identify potential use cases and Actors
15	Jaiwai, M. et al.	2017	Extracting UML class diagrams from software requirements in Thai using NLP	MDL	Parsing of Requirements into Nouns, Noun Phrases, Verbs, Verb Phrases, etc. to identify class diagrams
16	Gokyer, G. et al..	2008	Non-functional requirements to architectural concerns: MML and NLP at crossroads	CL	Classification of NFRs
17	Shreda, Q.A. et al.	2021	Identifying Non-functional Requirements from Unconstrained Documents using Natural Language Processing and Machine Learning Approaches	CL	Classification of NFRs
18	Vogelsang, A. et al.	2019	Supporting the development of cyber-physical systems with natural language processing: A report	CL	Differentiation in information and requirement
19	Gröpler, R. et al.	2021	NLP-based requirements formalization for automatic test case generation	CL	Entities are divided into syntactic and semantic (action, action constraints, subject/object) and (signal, attributes, actor/component)
20	Anwar, M.W. et al.	2020	A Natural Language Processing (NLP) Framework for Embedded Systems to Automatically Extract Verification Aspects from Textual Design Requirements	CL	Differentiation in Actions and Conditions
21	Sundararajan, M. et al.	2018	Requirements complexity definition and classification using natural language processing	CL	Classification of requirements according to complexity
22	Wang, H. et al.	2020	Improving efficiency of customer requirements classification on autonomous vehicle by natural language processing	CL	Classification of customer requirements: "environmental, energy, costs", function, perception, privacy, security, safety and "social, legal, ethical".
23	Varenov, V. et al.	2021	Security Requirements Classification into Groups Using NLP Transformers	CL	Classification of requirements in safety classes: Confidentiality, Integrity, Availability, Accountability, Operational, Access control, ...
24	Koscinski, V. et al.	2021	A Natural Language Processing Technique for Formalization of Systems Requirement Specifications	CL	Classification of requirements in Subject, Relation, Object, Descriptor, Case, Negation
25	Allala, S.C. et al.	2019	Towards transforming user requirements to test cases using MDE and NLP	CL	Classification of user requirements to derive test cases
26	Shehadeh, K. et al.	2021	Semi-Automated Classification of Arabic User Requirements into Functional and Non-Functional Requirements using NLP Tools	CL	FR and NFR using rules/heuristics

MDL: Modeling CL: Classification

Figure 5: Summary of all papers

https://doi.org/10.1017/pds.2023.277 Published online by Cambridge University Press