Automatic Information Extraction from Text-Based Requirements

Simon Fritz, Vethiga Srikanthan, Ryan Arbai, Chenwei Sun, Jivka Ovtcharova, and Hendro Wicaksono

***Abstract*—Requirements form the legal basis for many development pro-jects. They are usually exchanged between customer and supplier in the form of product and requirements specifications and re-quire a subsequent integration effort into the corresponding requirements management solutions. Especially for small and medium-sized enterprises (SME), which mainly use office solutions for the management of requirements, this involves a very high integration effort, which is why this is usually only partially managed or not managed at all. Software solutions available on the market already offer support, but they are too expensive or complex, especially for small companies.**

The project DAM4KMU, funded by German Federal Ministry for Education and Research (BMBF), addresses this challenge and by enabling SMEs from Germany to integrate requirement documents automatically into existing requirement structures with the help of NLP-based techniques. For this purpose, the documents to be processed are divided into semantic roles, which can then be transferred into a semantic data structure. This in turn enables an automatic linking of the requirements and system components, which reduces the manual effort and avoids possible errors.

***Index Terms*—Requirements engineering, NLP, context-sensitive assistance.**

I. Introduction

Requirements management and its methods are an underestimated discipline in project management, especially in small and medium-sized enterprises (SME). However, this discipline can be significantly responsible for the success or failure of a product development [1], [2]. As several studies have shown, office solutions are mainly used for the management of requirements today. However, due to their lack of connectivity and the possibility of collaboratively working on documents, they are unsuitable for managing and communicating requirements across all project participants [3].

Furthermore, established software solutions for requirements management usually require intensive training and therefore rather unsuitable for use by all project participants. Especially in small and medium-sized companies, there are also the personnel and financial resource limitations, which lead to requirements not being adequately managed [3].

In order to enable non-experts to write standardized and complete requirements without having to use training-intensive software solutions, new approaches are needed that actively support the user in documenting requirements and the associated context information. This challenge is addressed by the project DAM4KMU – *Digital Assistant for Requirements Management for agile product development in small and medium-sized enterprises*, funded by the German Federal Ministry of Education and Research (BMBF).

II. State of the Art

A. Requirements Management

Requirements management is very important for today's SMEs. Regardless of the development phase, requirements management should not be limited in time. Through regular changes and corrections of customers and suppliers, but also through laws, standards and guidelines, requirements management must be continuously connected to the development process. From ambiguous and possibly contradictory formulations, unambiguous requirements must be gradually formulated in consultation with the customer. The earlier unambiguous requirements are formulated, the more costs can be saved (see Fig. 1).

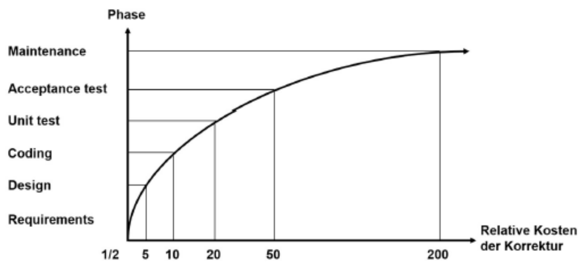


Fig. 1. Costs for the correction of requirements [4].

According to the Chaos Report of the Standish Group, 19% of all software projects failed, and in 52% of the cases the goals were not achieved [5]. The main reason for failure are insufficient processes, unclear responsibilities, insufficient requirements and poor handling of changing requirements [6]. A promising integration of requirements management has a positive effect on increasing productivity, reducing costs, shortening lead times and value orientation. Overall, the above-mentioned advantages can lead to an increase in efficiency of 20 – 40% of processes, with a high

Manuscript received October 5, 2020; revised November 20, 2020.

Simon Fritz, Ryan Arbai, and Chenwei Sun are with the Intelligent Systems and Production Engineering (ISPE), Research Center for Information Technology (FZI), Karlsruhe, Germany (e-mail: fritz@fzi.de, rarbai@fzi.de, sun@fzi.de).

Vethiga Srikanthan is with the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany (e-mail: vethisri@gmail.com).

Jivka Ovtcharova is with the Institute for Information Management in Engineering (IMI), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany (e-mail: jivka.ovtcharova@kit.edu).

Hendro Wicaksono is with the Industrial Engineering Mathematics & Logistics, Jacobs University Bremen gGmbH, Bremen, Germany (e-mail: h.wicaksono@jacobs-university.de).

doi: 10.18178/ijke.2021.7.1.134

probability of achieving project goals and avoiding missed deadlines and cost overruns [6], [7].

B. Boilerplates

A boilerplate, as shown in Fig. 2, assembles blocks in a given order, so that all the necessary information is available for a request and unnecessary information is excluded. The application can be used to ensure that all requirements are standardized and that an automated system, for example a link to related requirements, works more efficiently. In addition, it can be prevented that ambiguous requirements are formulated, which can lead to misunderstandings between a customer and supplier.

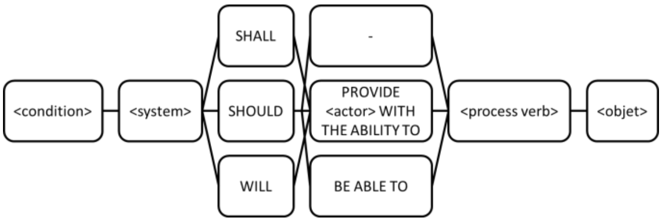


Fig. 2. Boilerplate for functional requirements.

Rupp [8] has put together different requirements templates for different additional information on a requirement. Each requirement has a priority or legal obligation, which can be expressed by the keywords used, also called priority words. According to Rupp there are three priority levels: Obligation, wish and intention. If the application of a requirement is mandatory, the word “shall” is used, for a wish “should” and an future intention “will” [8].

C. Boilerplates

To ensure the quality of requirements, more and more software solutions use NLP procedures. In the work of Jiang *et al.* [9] different NLP tools are compared with each other in terms of NER performance. Sanford NLP[1] and spaCy[2] stand out. At the Named Entity DATE spaCy does better than Stanford NER. Comparing the German Library of spaCy and Stanford CoreNLP, spaCy has more details. Therefore, spaCy is used in this paper.

SpaCy is an open source software library for NLP, written in the programming language Python and Cython. It provides neural network models for English, German, and several other European languages. It has its own functions for tokenization, lemmatization, named entity recognition, part-of-speech (POS) tagging and dependency parsing.

III. Related Work

The work of Fritz & Duecker [10] gives an overview of already existing areas of requirements management, application possibilities of already existing tools as well as possibilities for future tool functions. They focus on the development of assistance systems which offer a non-expert the possibility to manage requirements efficiently. They present two tools for requirements documentation: RQS and DESIRe. Both tools analyze a requirement in terms of completeness and offer suggestions for clear requirement formulations. Fritz and Duecker see a need for improvement in the avoidance of ambiguities and redundancies and point out the importance of the integration of external documents, as they are dealt with in this paper.

Vlas and Robinson [11] offer an automated technique for the detection and classification of requirements in natural language texts. Their development is a pattern-based approach, mainly based on grammar and layers. The first two layers use tokenization and part-of-speech tagging, the following three layers use logical statements, and the last layer classifies parts of the text based on the previous layers. The concept achieves a precision of 94% and a recall of 64%.

In the work of Waël *et al*. [12] a concept is developed to extract requirements from legal texts and regulations of a company. For the processing of the requirements they divide them into three types: Procedural instructions, explanatory statements and ontology statements. Based on these types, a UML model for classification is developed, which shall serve as a guideline for the translation of requirements into formal language.

Klaus *et al*. [13] are working on a solution called ReqCheck, which performs an automated quality check of natural language documents. As a basis, a software is used that checks free text according to linguistic rules, splits it into exam-relevant units and analyses them linguistically. From this software rules are extracted that apply to requirements, such as rules of unambiguity and comprehensibility. This approach achieves a precision of 97.05%, whereby the recall is not examined.

An ontology has also been used by Wicaksono and Schubert to model correlations between requirement items, product properties, product usage contexts, and customer profiles [14]. The correlations in the ontology are generated by analyzing the user feedbacks collected during product use phase [15]. The correlations are represented using SWRL rules. The rules indicate supporting and contradicting requirements [14].

The Reuse Company has developed four tools to facilitate requirements engineering: The RQA - Quality Studio checks for the CCC criteria (Correctness, Consistency, Completeness) of requirements. These requirements can be available both as text and as a model. With the KM - Knowledge Manager, ontologies can be created according to the subject area. This allows requirements to be completed with terms from the ontology. The RAT - Authoring Tool offers both the tasks of the RQA, as well as correction suggestions for requirements in real-time. Traceability Studio can be used to create connections between documents, requirements, or topics. Fraga and others also use an ontology approach [16]. They use requirements to build an ontology from them. With a semantic approach whole requirement are translated into a graph. Thus, different requirements can be checked for equality or contradiction. The graph can also be checked for incompleteness.

IV. Concept

Similar to the concepts described above, our concept uses

[1] https://nlp.stanford.edu/

[2] https://spacy.io/

computer linguistic methods for the processing and analysis of requirements texts. The present work additionally pursues the following goals:

- Automatic information extraction from textual requirements texts
- Automatic completeness check using boilerplates
- Automatic derivation of system components
- Automatic derivation of the hierarchical system structure of all system components
- Automatic linking of the requirements with the respective system components

With the help of these procedures, even non-experts should be enabled to integrate requirements documents into the existing requirements database without much effort. In addition, the automatic linking of all information with each other should improve the clarity and communication of new requirements to the respective system component managers. This is to avoid that new requirements are not considered or forgotten.

The present concept focuses on textual requirement descriptions. Diagrams such as UML or SysML are not considered, as initial investigations within the DAM4KMU project have identified a need for action, especially in the area of textual requirement descriptions.

In order to achieve the above-mentioned goals, the requirements document to be analyzed is first divided into individual sentences. Then each sentence is examined individually for the information it contains. Based on the sentence templates of SOPHIST [17], semantic roles are derived (see Table I) which are then used to automatically link the information.

TABLE I: SEMANTIC ROLES [INSPIRED BY, 17]

Semantic Roles	Description
Actor	Performing person (user) or object of an activity
Comparison operator	A comparison operator is a logical operator, which is applied to two arguments and returns a truth value
Object	Object which interacts with or is influenced by the reference object in any way. The respective interaction or influence is specified by the process word.
Parent-Object	Hierarchically superior object of the object
Parent-Subject	Hierarchically superior object of the subject
Priority word	Priority words, also calls key words, indicate the importance of a request
Process word	The process word specifies the manner of interaction of a condition or requirement
Property	A characteristic belonging to the essence of a reference object
Property Value	Value associated with a property consisting of a number or group of words and the appropriate unit
Subject	The subject is an object that is specified by a request or processed by a task

By automatically dividing the requirements to be processed into semantic roles, they can not only be used for completeness checks of the requirements as described in [16], but also for automatic derivation of the system structure. If, for example, requirements of a child element are specified, e.g. "the door of the car should...", a hierarchical dependency can be derived directly on the basis of the semantic role subject and parent-subject. The same applies to the parent-child relationship between objects. If this method is applied to all requirements, the structural relationships of the system can be derived and at the same time the associated requirements can be linked to them (see Fig. 3).

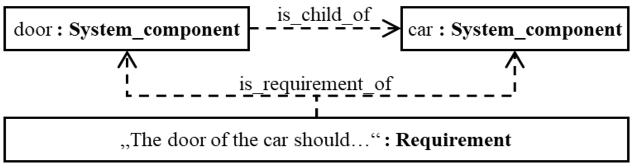


Fig. 3. Relations-Example.

This procedure enables the component owners to identify all requirements relevant to them without manually linking the requirements. As shown in this example, the system could automatically inform those responsible for the "door" and " car" components about the new requirement, thus avoiding possible mistakes.

Requirements of higher-level elements can also be inherited directly by the hierarchically subordinate elements, which reduces additional manual linking effort. For example, standards that apply to the system component “car” could be automatically passed on to the child element “door”, so that the person responsible for the component “door” can check it for relevance without either person having to create this link manually.

The semantic data structure of the present concept essentially consists of system components and requirements, which can be summarized as abstract elements of the class context element. System components in turn can also have properties, which in turn have a property value and a comparison Operator (e.g. “<”, “>” or “=”). Requests, on the other hand, should always be given a priority in order to know their importance.

In order to be able to link several context elements with each other, there is also a table of relations, each of which consists of two context elements and one relation type. With the help of the relation type, different types of relations can be mapped, whereby only the two types "is_child_of" and "is_requirement_of" are relevant for the present concept (see Fig. 4).

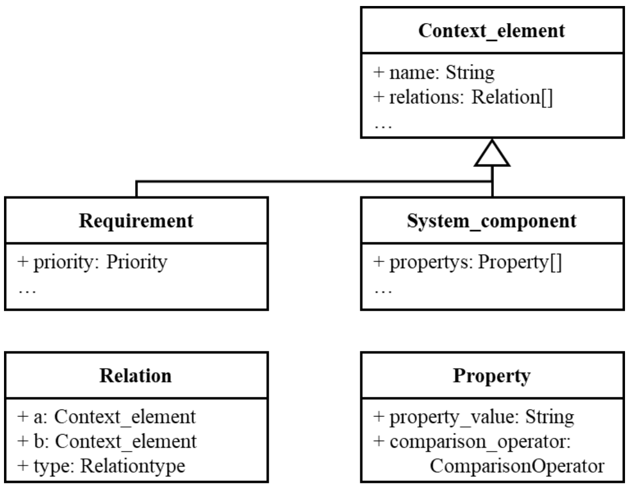


Fig. 4. Class diagram of the semantic data structure.

For use in an industrial context, these classes can be extended by further parameters (e.g., costs, status, technical drawings etc.), but this is not relevant for the present concept.

In order to check the completeness of requirements, this concept provides for the comparison of the recognized semantic roles with the boilerplates according to SOPHIST [17]. If, for example, no priority or subject is described within a requirement, the system can then instruct the user to add the missing semantic role. This procedure should reduce possible misunderstandings caused by incomplete sentences.

V. Implementation

Since requirements documents are mainly exchanged as PDF of MS-Office documents, the implementation python-docx for processing MS-Word documents was used for the first Proof of Concept.

To classify each sentence as requirement or non-requirement we use the TextCategorizer from spaCy and trained it with 600 sentences (300 requirements and 300 non-requirements) for 20 epochs.

For extracting the previously described semantic roles from textual requirements, the python library spaCy is used for this concept. For this purpose, linguistic rules were derived empirically, which assign their corresponding semantic role to the respective words or parts of sentences on the basis of Part of Speech (POS) and dependency (dep) tags. Named Entity Recognition (NER) was not used because too little training data was available to achieve meaningful results. However, the use of a NER model will be further evaluated within the DAM4KMU project. In Table II the linguistic rules per semantic role are listed:

TABLE II: Detection Rules for the Semantic Roles

Semantic Roles	Linguistic rules
Actor	POS: NOUN, PROPN Dep: ag, pnc or nk of the Object
Comparison operator	POS: ADV Dep: mo
Object	POS: NOUN, PROPN Dep: oa
Parent-Object	POS: NOUN, PROPN Dep: ag, pnc or nk of the Object
Parent-Subject	POS: NOUN, PROPN Dep: ag, pnc or nk of the Subject
Priority word	POS: VERB, VMFIN Dep: ROOT
Process word	POS: VERB, AUX Dep: oc, cj, mo
Property	POS: NOUN, PROPN Dep: - Each match is matched against a word list based on [18] or with the ending “-heit”, “-keit” or “-tät”.
Property Value	POS: ADJ, NUM Dep: cc, pd, ams or mo
Subject	POS: NOUN, PROPN Dep: sb or ROOT

VI. Evaluation

For testing the text classification and the semantic role ex-traction 300 sentences (100 functional and 50 non-functional requirements as well as 150 non-requirements) from different domains where used. To evaluate the procedures, the F1 score was chosen as a metric.

A. *Text Classification*

For the text classification a precision, recall and F1-Score of 94,6% has been achieved. Furthermore, spaCy's TextCate-gorizer needs an average of 190ms for 100 sentences on a conventional Lenovo P71 notebook. The accuracy and speed of this method allow the conclusion that it can also be used in an industrial context, but that a manual final check is neces-sary, to avoid the error of 5,4%.

B. *Extracting Semantic Roles*

Table III shows the precision, recall and the F1 Score of all semantic roles within the 150 requirements. For the extraction of semantic roles, the described procedure requires on average 1.19 seconds for 100 sentences with a Lenovo P71 notebook.

TABLE III: Evaluation of the Identification of the Semantic Roles

Semantic Roles	Precision	Recall	F1 Score
Actor	0,928	1,00	0,962
Comparison operator	0,391	0,409	0,400
Object	0,971	0,705	0,817
Parent-Object	0,982	0,882	0,913
Parent-Subject	0,952	0,92	0,932
Priority word	0,935	1,00	0,966
Process word	0,888	0,851	0,869
Property	1,00	0,451	0,622
Property Value	0,964	0,936	0,949
Subject	0,887	0,859	0,873
Average	0,890	0,801	0,830

As can be seen here the roles “Property” and especially “Comparison operator” are detected with a comparatively low precision.

The reason for the inaccuracy of detecting properties is mainly due to the linguistic similarity of properties and sub-jects e.g. "the door of the car" and "the color of the car". In both cases, "the door" and "the color" are recognized as sub-jects. In order to distinguish them from each other, extensive word lists are necessary, which must be adapted domain-specifically, which leads to the inaccuracy of the approach shown.

The linguistic rule for " Comparison operator" therefore does not work very well, because besides adverbs many special characters can be used, such as ">", "<" or "=". A word list could also help here. However, since even properties are only recognized correctly with low probability, a NER model based on Sentence Transformers is to be evaluated within the DAM4KMU project. The consortium hopes to achieve better results even with comparatively low training data input.

Since parent elements can be recognized with an accuracy of 91.3%, the procedure should be able to automatically extract and link the system structure described within a requirements document. For practical use, a manual final check should also be provided to correct possible errors.

C. *Completeness Check*

As shown above, many of the semantic roles with high hit probability are correctly identified, which is why completeness can also be checked correctly with the corresponding probability. However, due to the poor F1 score of properties and comparison operators, non-functional requirements in particular are not checked correctly. If the previously described planned improvements of the semantic role recognition can be implemented within the project DAM4KMU, the automatic completeness check will also be usefully applicable. However, at this point in time, this is not yet practicable, as initial tests show, which is why it was not further evaluated at this point.

D. Completeness Check

The most advanced software solution for the administration and management of requirements is the Reuse Company solution. With the help of sentence templates and extensive word lists, which must however be adapted for the respective do-main, requirements can be automatically recognized and checked for completeness.

The present concept addresses mainly small and medium-sized companies that cannot or do not want to afford an ontology that is specially adapted for them, as it is necessary for the Reuse Company solution. With its rule-based approach, the present concept is intended to provide a domain-independent solution, which is not yet fully possible at this point in time due to the partial use of word lists that is necessary. In the context of the DAM4KMU project, the use of pre-trained language models such as BERT for the recognition of semantic roles using an adapted NER model will therefore be evaluated in subsequent work.

In addition to the rule-based role recognition, the present concept also offers the automatic linking of information and the automatic generation of system graphs, which sets it apart from current solutions available on the market. Although the Reuse Company's solution offers corresponding filter options, it does not allow for a hierarchical system component structure, which is why requirements have to be assigned manually to the persons responsible.

However, the present concept shows the general feasibility of the defined goals and offers sufficient starting points for further assistance systems based on semantic role recognition.

VII. Conclusion and Outlook

In the context of this thesis a concept for the automatic ex-traction and linking of requirements and the related system components is presented. With the help of this procedure, requirements documents can be transferred automatically into already existing requirements databases with little effort. For this purpose, requirements are first separated from non-requirements. Afterwards all system components, which are described within the textual requirements, are extracted and linked together. This creates a system hierarchy, which makes it easier for the responsible persons to identify the requirements relevant to them.

The prototypical realization of this concept is based on the Python libraries python-docx and spaCy. With the help of python-docx the texts contained in Office documents are extracted and made available for further processing. Afterwards, a text classification based on spaCy is used to separate requirements from non-requirements. So-called semantic roles are extracted from the requests, which represent specific in-formation within the sentences. Based on these semantic roles, a system component hierarchy can be derived automatically, which is linked to the respective requirements. In addition, the semantic roles and user-defined boilerplates can be used to check the requirements for completeness in order to reduce possible errors.

The presented text classification already achieves an F1 score of 94.6% with comparatively few training data (600 records). With the purely rule-based recognition of semantic roles, the roles "Property" and "Comparison operator" are not recognized sufficiently well (F1-Score < 80%). For this reason, NER models will be developed in the further course of the DAM4KMU project, which are based on pre-trained language models (such as BERT) in order to increase the accuracy and to ensure a non-domain-specific transferability of the concept.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

SF developed the main concept and wrote the paper; VS analyzed the data; RA implemented the prototype; CS helped writing the paper; JO and HW reviewed the paper; all authors had approved the final version.

References

[1] The Standisch Group. The CHAOS Report. (1994). [Online]. Available: https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf

[2] The Standisch Group, Chaos Report: 21st Anniversary Edition. [Online]. Available: https://www.standishgroup.com/sample_research_files/CHAOSReport2014.pdf

[3] S. Adam, S. Wünch, and N. Seyff. (2014). Ergebnisbericht 2014/15 RE-Kompass. [Online]. Available: https://www.hood-group.com/fileadmin/user_upload/requirements/Downloads/Paper/DE/HOOD_RE-Kompass_2014_15_v1.6.pdf

[4] B. Schienmann, *Kontinuierliches Anforderungsmanagement: Prozesse, Techniken, Werkzeuge [mit Muster für Lasten- und Pflichtenheft]*, München: Addison-Wesley, 2002.

[5] S. Hastie and S. Wojewoda, Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch, 2015.

[6] C. Ebert, *Systematisches Requirements Engineering: Anforderungen Ermitteln, Spezifizieren, Analysieren Und Verwalten*, 5th ed. Heidelberg: dpunkt.verl., 2014.

[7] S. Fritz, W. Felix, and J. Ovtcharova. (2020). A guideline for the requirements engineering process of SMEs regarding to the development of CPS. [Online]. Available: https://ieeexplore.ieee.org/document/8710732

[8] C. Rupp, *Requirements-Engineering Und-Management: Aus der Praxis Von Klassisch Bis Agil*, 6th ed. München: Hanser, 2014.

[9] R. Jiang, R. E. Banchs, and H. Li, "Evaluating and combining name entity recognition systems," in *Proc. the Sixth Named Entity Workshop*, Berlin, Germany, 2016, pp. 21–27.

[10] S. Fritz and J. Dücker, "Notwendigkeit von assistenzsystemen zur durchgängigen digitalisierung des anforderungsmanagement," *ZWF*, vol. 113, no. 6, pp. 359–362, 2018.

[11] R. E. Vlas and W. N. Robinson, "Two rule-based natural language strategies for requirements discovery and classification in open source software development projects," *Journal of Management Information Systems*, vol. 28, no. 4, pp. 11–38, 2012.

[12] Second International Workshop on Requirements Engineering and Law, IEEE, 2009.

[13] A. Klaus, M. Koch, U. Reuther, and G. Hensberg. ReqCheck: Automatisierte Qualitätssicherung von Anforderungsdokumenten mit Hilfe linguistischer Regeln. [Online]. Available: https://pdfs.semanticscholar.org/1527/5d84ed24e6a8a2cde63b7667e6890a3b3aeb.pdf?_ga=2.145563191.2038943660.1581882233-327088078.1573337679

[14] H. Wicaksono, V. Schubert, S. Rogalski, Y. Ait Laydi, and J. Ovtcharova, *Enabling Manufacturing Competitiveness and Economic Sustainability: Ontology-driven Requirements Elicitation in Product Configuration Systems*, 2011, pp. 63-67.

[15] V. Schubert, H. Wicaksono, and S. Rogalski, *Global Product Development: A Product Life-Cycle Oriented Approach for Knowledge-Based Product Configuration System*, 2010, pp. 357-362.
[16] A. Fraga, J. Llorens, L. Alonso, and J. M. Fuentes, "Ontology-assisted systems engineering process with focus in the requirements engineering process," in *Proc. the Fifth International Conference on Complex Systems Design & Management CSD&M 2014*, 2015, pp. 149-161.
[17] SOPHIST GmbH. (2020). Schablonen für alle Fälle: Die SOPHISTen. [Online]. Available: https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/Wissen_for_free/MASTeR_Broschuere_3-Auflage_interaktiv.pdf
[18] U. Tellmann. Nominalisierung – Nomen aus Verben/Auswahlliste. [Online]. Available: https://www.wirtschaftsdeutsch.de/lehrmaterialien/grammatik-nominalisierung-liste-verben.pdf

Simon Fritz was born in Buehl, Germany on January 10, 1989. He has received his master of science in engineering with the following specializations product development and mechatronics at the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, 2015.

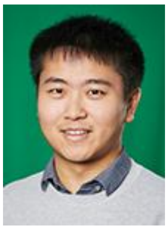
He worked as a researcher at the Research Center for Information Technology (FZI) for 5 years and works now as a project manager at the NETSYNO Software GmbH in Karlsruhe, Germany with the focus on process digitization, requirements engineering and natural language processing.

Vethiga Srikanthan was born in Kirchheim/Teck, Germany. She did her bachelor's degree in computer science at the University of Stuttgart, Stuttgart, Germany and her master's degree in computer science in 2019 with the specialization requirements engineering at the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany.

Today she is working as a systems engineer at ITK Engineering GmbH, Stuttgart with the focus on requirements elicitation.

Ryan Arbai is originally from Indonesia and studied computer science at Karlsruhe Institute of Technology (KIT). During his studies, he worked as a student assistant at Forschungszentrum Informatik (FZI) for 3 years.

He is now working as a software developer at NETSYNO GmbH.



Chenwei Sun was born in Hefei, China on January 25, 1993. He has received his master of science in engineering with the following specializations automotive technology and information technology at the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, 2019.

He has been working as a researcher at the Research Center for Information Technology (FZI) for a year.

Jivka Ovtcharova is a pioneer in virtual and lifecycle engineering, artificial intelligence and mixed reality. She is the head of the Institute for Information Management in Engineering at the Karlsruhe Institute of Technology (KIT) and director at the Karlsruhe Research Center for Information Technology (FZI), focusing on cutting-edge and human-centred lifecycle solutions for research and practice. The professor with a double doctorate in mechanical engineering and computer science is one of the 25 Women for the Digital Future in Germany, winner of the first Inspiring Fifty DACH Award 2019, internationally recognised expert, keynote speaker and supervisory board member.

Hendro Wicaksono is a professor of industrial engineering at Jacobs University Bremen, Germany. He has received his doctoral degree in mechanical engineering from Karlsruhe Institute of Technology. He has earned his M.Sc. in information and communication engineering from the University of Karlsruhe and B.Sc. in computer science from the Institut Teknologi Bandung (ITB), Indonesia. Before he joined Jacobs University Bremen, he worked in several companies as an IT consultant and as a researcher at FZI Research Center for Informatics Karlsruhe and Karlsruhe Institute of Technology. His research focuses on data management and analytics and their applications in supply chain management, smart cities, and sustainability.