



# On identifying possible artificial intelligence applications in requirements engineering processes

Simon Dehn<sup>1</sup> · Georg Jacobs<sup>1</sup> · Thilo Zerwas<sup>1</sup> · Joerg Berroth<sup>1</sup> · Matthis Hötter<sup>2</sup> · Matthias Korten<sup>2</sup> · Marvin Müller<sup>3</sup> · Nico Gossen<sup>3</sup> · Serena Striegel<sup>4</sup> · Dirk Fleischer<sup>4</sup>

Received: 15 November 2022 / Accepted: 24 February 2023 / Published online: 21 March 2023  
© The Author(s) 2023

## Abstract

Managing an ever-growing number of requirements is a time consuming and highly complex activity in product development projects. With the rise of Artificial Intelligence (AI) in the last decades, algorithms are now capable of supporting Requirements Engineering (RE) tasks. The algorithm's ability to intelligently process natural language data as well as the application in RE have been published extensively. In established RE processes however, it is often unclear, where AI algorithms can be utilized. The result is a significant imbalance between state-of-the-art AI algorithms and their application rates in real RE processes. One of the reasons is, that current models of RE processes aren't able to convey the needed information for the identification of suitable tasks. Therefore, this paper aims to provide a framework of finite and standardized process steps with a data processing-oriented perspective, that can be used to systematically identify the points in the RE processes where AI algorithms can be applied. With the standardized process steps of this framework, established RE processes can be abstracted and expressed in a way that is compatible with the scope of AI algorithms. Due to the finite number of standardized process steps, libraries of suitable AI algorithms for the defined process steps can be built, so that their applicability can be efficiently assessed. The presented framework was developed collaboratively during a research project with industry participation.

## Identifikation der Anwendungsmöglichkeiten von KI-Algorithmen in Requirements Engineering Prozessen

### Zusammenfassung

Das Management einer immer größer werdenden Menge an Anforderungen ist eine zeitlich aufwendige und komplexe Aufgabe in der Produktentwicklung. Durch den technologischen Fortschritt auf dem Gebiet der künstlichen Intelligenz (KI) sind Algorithmen heute in der Lage, einzelne Tätigkeiten innerhalb des Anforderungsmanagements (AM) zu unterstützen. Die Fähigkeit der KI, natürlichsprachliche Texte intelligent zu verarbeiten sowie ihre Anwendung im AM wurden schon mehrfach in diversen Veröffentlichungen gezeigt. Zwischen dem Stand der Forschung und der industriellen Anwendung von KI-Algorithmen im AM besteht jedoch ein signifikantes Ungleichgewicht. Ein Grund dafür ist, dass es in bereits etablierten AM-Prozessen häufig unklar ist, welche Prozessschritte durch den Einsatz eines KI-Algorithmus unterstützt werden können. In den gängigen Prozessmodellen des AMs fehlt es an Informationen, die für die Identifikation von KI-Einsatzmöglichkeiten notwendig sind. In diesem Beitrag wird ein Baukasten für standardisierte Prozessschritte vorgestellt, mit denen sich AM-Prozesse aus datenverarbeitender Sicht modellieren lassen, um so die systematische Identifikation von KI-Einsatzmöglichkeiten zu ermöglichen. Der Prozessbaukasten wurde im Kontext eines Forschungsprojektes und in Kollaboration mit Industriepartnern entwickelt.

✉ Simon Dehn  
simon.dehn@imse.rwth-aachen.de

<sup>1</sup> Institute for Machine Elements and Systems Engineering,  
Schinkelstr. 10, 52062 Aachen, Germany

<sup>2</sup> Institute for Automotive Engineering,  
Steinbachstr. 7, 52074 Aachen, Germany

<sup>3</sup> PEM Motion GmbH, Bohr 12, 52072 Aachen, Germany

<sup>4</sup> BMW AG, Petuelring 130, 80788 Munich, Germany

# 1 Introduction

Requirements provide the guideline for the development process and restrict the solution space for the design of systems [1, 2]. For successful and efficient developing, requirements not only need to be demand-oriented and precise, but also consistent and redundancy-free [3, 4]. Modern cyber-physical systems are driven by high customer expectations as well as strict regulatory frameworks, leading to an ever-rising number of requirements that need to be handled by developers [5]. The activities that are related to the work with requirements are summarized by the term Requirements Engineering (RE). In an industrial context, requirements are traditionally captured in the form of written text [6]. Therefore, most tasks in RE require the interpretation of natural language and are consequently hardly automatable with traditional computer algorithms. In addition, the individual steps that are carried out to solve RE related tasks are highly diverse, depending on the system under development and the perspective of the different development teams [7]. Consequently, RE ties up large amounts of human workforce capacities in heterogeneous RE processes, which are increasingly reaching their limits in view of the further growing requirement volumes and technical complexity of cyber physical systems. In order to manage complexity in systems, Model-Based Systems Engineering (MBSE) approaches, e.g. [8–11], offer a function-oriented and seamless connection of digital models, allowing for automating development tasks to some extent. Manually creating such digital models of already existing textual requirements in large scale however, admittedly comes with great initial efforts.

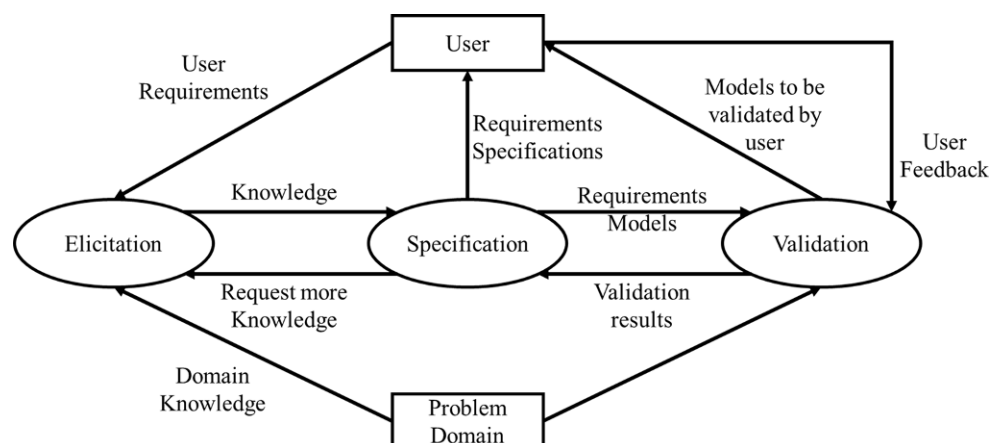
Advances in the field of Artificial Intelligence (AI) offer an alternative approach to support RE tasks. Natural language processing (NLP) algorithms in particular are now able to interpret and process presently existing textual data [12], providing a promising approach to support RE tasks. As recent literature shows however, there still is a large

gap between the state-of-the-art and industry application of AI algorithms [6, 12, 13]. To broadly apply AI methods in RE processes, it is necessary to systematically identify the activities in existing RE processes that are suitable for the application of AI.

In order to close this gap, we present a set of standardized process steps that can be used to conduct a systematic identification of AI application opportunities in RE. We define elementary process steps (EPS) that enable the abstraction of individual process steps within RE processes by representing them as standardized input-output models. The term *elementary* refers to the scope of the process steps: Analogous to AI algorithms, that are typically trained to perform a concrete task, EPSs also describe a specific and elementary task, that cannot be further decomposed in any meaningful way. This allows for a precise and homogeneous description of RE processes. Based on the input-output models and the same degree of abstraction, capable AI methods can then be linked to the corresponding EPS. Ultimately, this allows for a systematic identification of AI application opportunities in each RE process modeled with EPS. This approach was developed based on real RE processes in a collaborative research project with participation of the automotive industry.

The paper is structured as follows. Chapter 1 gives a brief introduction into the topic and aims to outline the problem. Chapter 2 summarizes the state of the art in RE as well as AI in RE. Chapter 3 follows with the research question that is derived from the state of the art. Chapter 4 describes the methodology used to conceive the EPS. An exemplary application of the EPS is presented in Chap. 5. Chapter 6 closes this paper with a conclusion and an outlook for future work.

**Fig. 1** The iterative RE process model according to [20]



## 2 State of the art

In literature, the term Requirements Engineering (RE) is used ambiguously [14]. In this paper, the following definition for RE by [15] will be used: “The subset of systems engineering concerned with discovering, developing, tracing, analyzing, qualifying, communicating and managing requirements that define the system at successive levels of abstraction.” This chapter provides relevant, available information on RE processes as well as AI applications in RE.

### 2.1 Requirements engineering processes

In RE, there is no standard process model that is able to grasp the highly individual approaches that exist in reality [16]. In order to be able to match existing AI algorithms to RE process steps, the process model should fulfill two criteria: First, the process model should specify RE process steps on a level that matches the scope of AI algorithms. Second, it is important that the process model relates to typical requirement data transformations, e.g. by specifying relevant data inputs and outputs. In [17] alone, eight different RE process models are presented. All of these RE process models describe the individual RE process steps only from a business process perspective, that makes it impossible to match suitable AI algorithms. The linear RE process model by [18] for example, divides the RE process into five consecutive steps described on a unspecific level, e.g. *analysis and modelling*. No relation to the processed requirement data is given and no inputs and outputs of the individual steps are defined. The same conclusions can be made for the spiral model by [19] as well as the iterative RE process model by [20]. Although the latter process model roughly defines inputs and outputs for the process steps, Fig. 1 shows, that there is no clear reference to transformed data artifacts.

Another commonly mentioned RE process model is the ISO standard 29148 [4]. Here, individual RE process steps are elaboratively described, but neither is a relation to actual requirements data transformation apparent, nor are the task definitions intended to be within the scope of algorithm capabilities. What stands out however, is the occurrence of similar phases, that are repeatedly used in most of the different process models. The re-occurring RE phases, that are also mentioned in [6, 18, 21], can be described as follows:

1. *Requirements Elicitation*. This phase describes the process of gathering all of the sources that possibly contain information for the system under development and must be translated into requirements. Sources can for example include stakeholder surveys, ISO standards, interna-

tional and regional regulations or internal requirements that have been used in previous product cycles.

2. *Requirements Documentation*. The requirements documentation phase describes the process of transforming the gathered information of the previous phase into a formal representation that is suitable for the developing process. Typically, the output of this phase is a collection of requirements in the form of textual natural language. More sophisticated formalizations incorporate predefined sentence structures [22] or use additional attributes along with the requirements' text. Beside the textual representations, requirements can also take other forms, e.g. a Use-Case model in UML [23].
3. *Requirements Analysis*. This phase considers a variety of activities that include the formal analysis of the documented requirements, i.e. confirming that formalization rules have been met, but also analysis in regards to the content of the requirement, e.g. finding inconsistencies, ambiguities or dependencies.
4. *Requirements Verification*. Tasks in this phase entail the examination of development artifacts such as software code, system architectures or simulation results to ensure the fulfillment of all specified requirements.

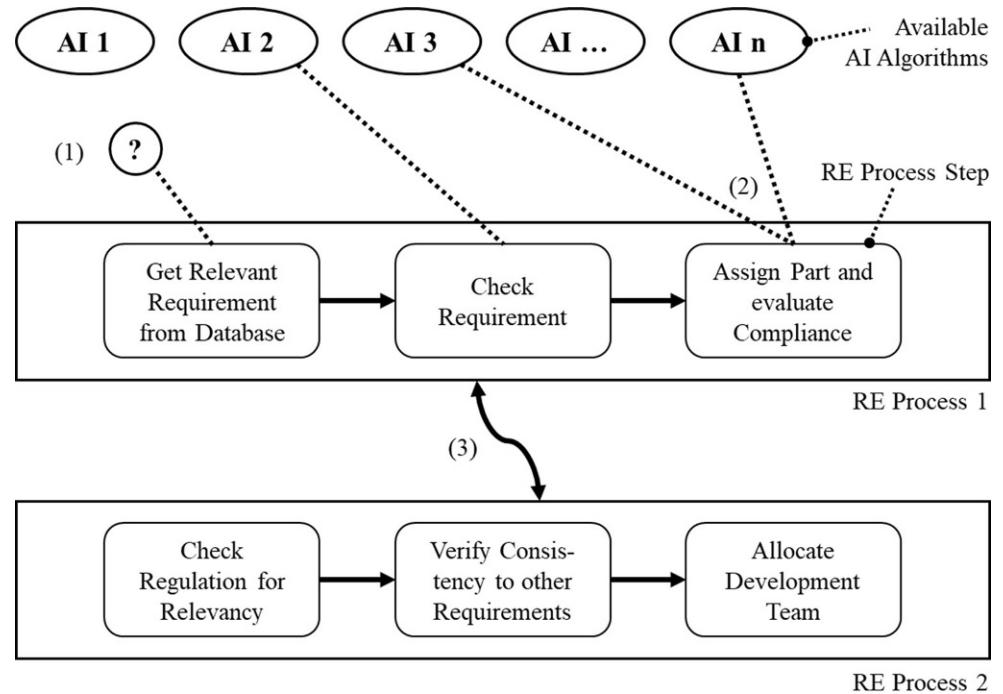
### 2.2 Artificial intelligence in requirements engineering

The application of AI technologies in RE is a promising approach [24]. With the advance of natural language processing (NLP) technologies, AI algorithms have become capable of interpreting and processing textual data. In a systematic literature research, [25] classifies typical AI capabilities to support RE processes as follows:

1. *Detection*. Identifying linguistic issues in requirements text.
2. *Extraction*. Finding domain-typical terms in requirements, e.g. to create glossaries.
3. *Classification*. Classifying requirements according to predefined types, e.g. functional and non-functional.
4. *Modelling*. Generation of requirements models.
5. *Tracing & Relating*. Identifying correlations between requirements.
6. *Search & Retrieval*. Algorithms of this type can be used to find single or multiple requirements out of requirements repositories.

In total, over 350 different AI algorithms are classified by [25] and cross-referenced to the RE phases presented in 2.1. The survey concludes, that most of the AI algorithms are developed with the capability of *Detection* and used in the *Requirements Analysis* phase. However, only 7% of all algorithms are evaluated in an industrial context. The authors also show, that the AI algorithms' capabilities are used

**Fig. 2** Shortcomings of surveyed RE process models for allocating AI algorithms



in more than one of the RE phases. AI algorithms, that are capable of *Detection* for example, are also used in other RE phases such as *Requirements Elicitation* and *Requirements Verification*. This hints at reoccurring tasks being carried out across different RE phases. Yet, it is still unclear which AI algorithm can be applied for which particular process step.

### 3 Problem statement and research need

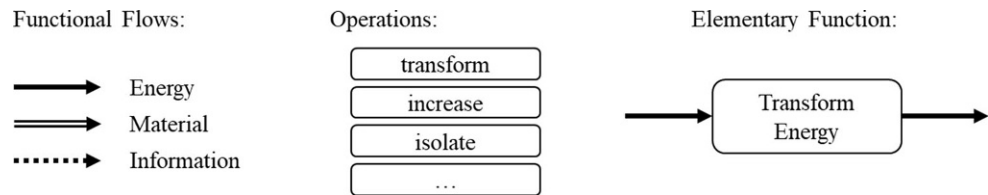
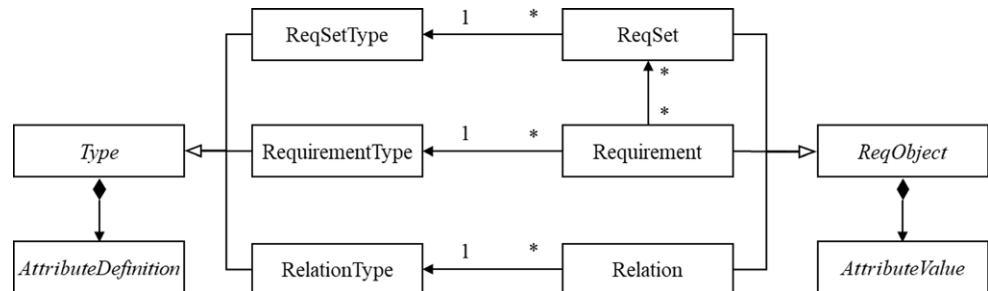
As shown in Chap. 2, a multitude of AI applications have been proposed for tasks within the RE phases [6, 25]. However, there is no systematic way of transferring AI technologies into practice [26] of RE. The result is a gap between the state-of-the-art in AI capabilities for RE and the actual application in industrial contexts [25]. As stated by [13], this is due to the lack of a systematic approach of identifying the tasks that can be supported by AI algorithms. Moreover, as stated by [24], it is often not clear what RE task can be supported by a newly developed AI algorithm.

From the reviewed RE process models, it is apparent, that the commonly used RE phase descriptions are not suitable to allow assessment of algorithm deployment opportunities, as none of them meets the two criteria stated in Chap. 2. Algorithms are usually built to process input data into output data [27]. But since RE processes are so highly individual on this level of specificity, a generic, detailed process model for all relevant RE process steps is described as impossible by [28]. This individuality also became evident in a survey about RE processes in six different development teams from

the automotive industry, that has been conducted by the authors. In particular, three shortcomings in the modeling of RE processes were perceived in the survey. In reference to Fig. 2, the shortcomings of the RE process models for identifying AI algorithm applicability are the following:

1. No reference to the processed RE-related data is recognizable.  
In particular, this shortcoming describes the inability to allocate an AI algorithm, because of missing information about the process step: Which inputs, i.e. RE-related data artifacts, are processed into which outputs by an individual process step.
2. The scope of the process steps described in process models varies and is mostly too broad for matching a single AI algorithm.  
In this case, an unambiguous allocation of a single AI algorithm is not possible, because the individual process step is described on a level that is too general and consequently has to be split up or defined more precisely.
3. There is no uniform representation for RE processes that allows for an accurate allocation of AI algorithms.  
This shortcoming becomes apparent when considering more than one RE process: When evaluating RE process 2 for AI applicability, as depicted in Fig. 2, the allocation of AI algorithms to RE process steps must be conducted again. The knowledge about suitable allocations of AI algorithms from RE process 1 is lost and can't be reused for application in RE process 2.

In order to address these shortcomings, the following research question is posed:

**Fig. 3** Concept of Elementary Functions according to [29]**Fig. 4** A simplified requirements data meta-model of the ReqIF standard [30]

How can highly individual RE processes be homogeneously represented, such that the processed requirement data is apparent and suitable AI algorithms can be systematically allocated?

To guide the research activity and narrow the solution space, the following hypotheses are derived from the presented shortcomings.

1. The inputs and outputs of the process steps can be utilized to relate to typical requirements data artifacts.
2. The process steps can be defined on an elementary level, such that they are generally applicable and represent RE process steps in the scope of AI algorithms.
3. RE processes can be represented by a finite set of standardized process steps, allowing for an unambiguous allocation of AI algorithms

In response to the research question and along the hypotheses a framework of elementary process steps was developed. The derivation of the framework is outlined in Chap. 4.

## 4 Conceiving the EPS framework

In this chapter, the conceiving of the Elementary Process Step (EPS) framework is described. Instead of proclaiming a one-size-fits-all RE process model, the authors of this paper propose using a set of preliminarily defined and generally applicable elementary process steps, that represent single data-oriented tasks and can be combined to express any individual RE process. The key aspects of this framework are based on a methodical approach that has been successfully used in early systems engineering works by [29]. It is important to note, that [29] describes a method, that is intended to model system functions and is explic-

itly no process modeling technique. Albeit being used for a different context, said approach is chosen as an analogy, because it offers a solution to a problem with similar characteristics. System functions, as well as RE process steps, are naturally expressed very individually and can be specified in a wide range of scopes. To homogenize system functions, the authors of [29] define a finite set of universally applicable *elementary* functions in the form of input-output models. This is achieved by defining the possible in- and outputs, i.e. the *functional flows*, as well as the operations that can be performed on the functional flows. In combination, the functional flows and the operations form the set of elementary functions as depicted in Fig. 3.

The term *elementary* refers to the scope of the functional description. The scope is chosen, so that an allocation of individual physical effects that are able to realize each elementary function can be made. In summary, this approach enables a standardized representation of functions with explicit reference to the type of the processed entity, along with allocated solution proposals in the form of physical effects.

Analogically, the EPS framework aims to define a set of *elementary* process steps, that enable a standardized representation of highly individual RE processes. Here, *Elementary* refers to the scope that is chosen, so that it fits the scope of AI algorithms. The AI algorithms in turn enable the task described by the individual EPSs. In order to conceive such a framework, the possible inputs and outputs, i.e. the *RE flow types*, as well as the employable operations, i.e. the *RE operations*, have to be defined.

### 4.1 Conceiving the RE flow types

In the following, the possible inputs and outputs, i.e. the RE flow types, of the EPS framework are defined in analogy to



**Table 1** Definition of RE Flow types

RE Flow Type	Flow Component	Definition
Requirement Set	Format Content	A collection of requirements
Requirement	Format Content	An individual requirement
Relation	Format Content	Association between data artifacts
Information	–	Any external data artifact

the functional flow types *Energy*, *Material* and *Information*. According to [27], a computer algorithm is characterized by its capability to transform specified inputs into specified outputs. In accordance to hypothesis 1 as defined in Chap. 3, an input-output representation of the performed process step is therefore generally applicable. For the consideration of the inputs and outputs and to determine the relevant data artifacts the algorithms should process, a simplified version of the ReqIF [30] meta-model is used. This meta-model is shown in Fig. 4 using the UML class diagram notation [23].

The meta-model depicts the main requirement data artifacts and their relations to each other. The *ReqSet* represents a collection of individual *Requirements*. A *Relation* is used to express links between data artifacts. All of the three entities *ReqSet*, *Requirement* and *Relation*, are associated with the superclass *ReqObject* and can contain *AttributeValues*. The definition of the attributes for each *ReqObject* is specified by the corresponding *Type* or the contained *AttributeDefinition* respectively. To summarize, the derived RE flow types that can be defined for the EPS are *Requirement Set*, *Requirement* and *Relation*. According to their type, each of the three flow types has an *AttributeDefinition*, which will be referred to as the *format*, and corresponding *AttributeValues*, which will be referred to as the *content*. In order to enable the modeling of interactions with external data artifacts, a fourth flow type, *Information*, is added to the definition of the RE flow types. The *Information* flow type will be used to express data artifacts that are relevant in RE and processed by the process steps, but are not directly contained in the requirement data artifacts, e.g. technical drawings, software specifications or user reviews. Table 1 shows the derived RE flow types and their definitions.

**Table 2** Definition of the RE Operations

RE Operation	Definition
Modify	Creation, deletion or editing of data artifacts
Verify	Any examination of data artifacts
Extract	Extracting of relevant information from data artifacts

## 4.2 Conceiving the RE operations

For the definition of the operations, that can be performed on the RE flow types, a survey of five extensive RE processes, containing up to 75 individual process steps from different departments within an automobile company, have been conducted and analyzed. Additionally, a cross-examination of the task and phase definitions given by literature [15, 25, 31] against the surveyed processes was carried out, in order to achieve a representative basis for the derivation of the RE operations.

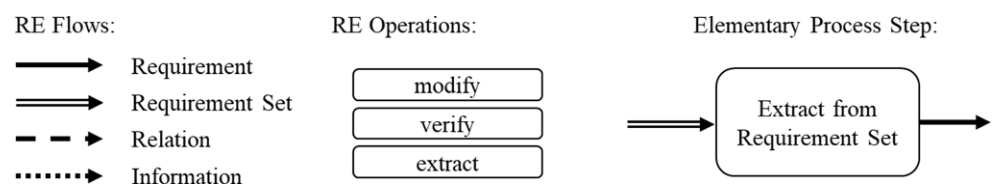
For example, the RE operation modify captures all kinds data artifact modification including changing or further detailing of a single requirement. To keep the amount of operations limited and in contrast to the definition given in Chap. 2, only those operations, that interact with requirement data artifacts are included. For example, the mere act of saving requirements to a database or sending requirements specifications via e-mail are intentionally left out. The resulting RE operations are shown in Table 2.

## 4.3 The resulting EPS framework

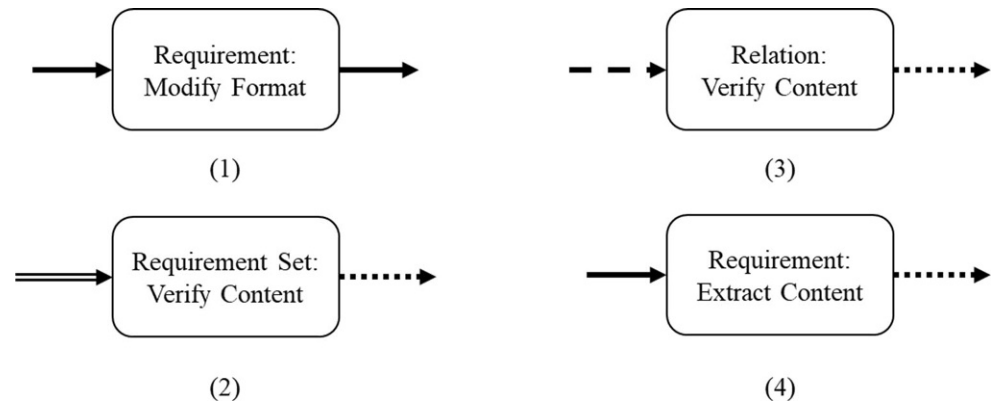
The resulting EPS framework is depicted in Fig. 5. The framework consists of the three RE operations *modify*, *verify* and *extract* as well as the four RE flow types *Requirement Set*, *Requirement*, *Relation* and *Information*. The individual EPS are formed by combining the RE flow types with the RE operations.

In accordance to the meta-model described in Chap. 4.1, the RE flow types are differentiated further into *format* and *content*. Figure 6 shows four exemplary EPSs. EPS (1) describes the modification of the format of a requirement. The content of the requirement itself is not affected. In practice, this task emerges, when transferring a textually formatted requirement, e.g. from a regulatory document, into a structured data format within a database of requirements containing supplementary attributes. EPS (2) describes the

**Fig. 5** The EPS Framework, consisting of the RE flow types and RE operations that can be combined to form the elementary process steps



**Fig. 6** Exemplary EPS, formed by the combination of RE flow types and RE operation



verification of a requirement set. When developing a system, the set of the demanded requirements that are included in a requirements specification, need to be consistent. So, for example, this EPS is used to describe a consistency check of the requirements stated in a requirements specification. The output is of the type *information*, as it provides the result of the verification. EPS (3) can be used to express the verification of a relation between two data artifacts. The type of the relation is specified in the format of the relation. For example, it can depict the verification of a *satisfy* relationship between a requirement and a specified subsystem. EPS (4) uses the *content* of a single requirement as an input, performs an extraction on the content and provides an *information* output. This EPS can therefore describe the extraction of relevant information out of the textual description of one requirement, e.g. acceptance criteria for testing purposes.

The total number of resulting EPS is 18. This is given by the combinatorial set of RE operations and main RE flow types *Requirement*, *Requirement Set* and *Relation*, including their further differentiation of *format* and *content*. The *Information* flow type is explicitly left out of the computation of the combinatorial set, because it acts as a secondary flow, that can be added optionally. The EPS have been documented in a catalog as shown in Fig. 7.

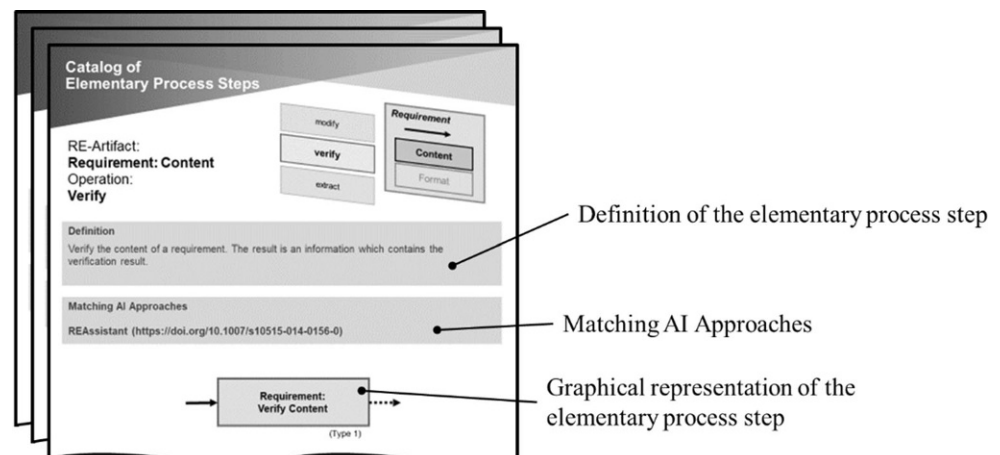
For every EPS, a textual definition and graphical representation is given. Wherever possible, matching AI approaches are linked. The EPS can be used to model actual RE processes, while closely following the data-operations that are performed on the requirements. The modeled process can then be assessed in terms of time, quality and cost for each process step. For critical process steps, the applicability of AI algorithms can then be checked using the corresponding EPS catalog page.

## 5 Application of the EPS framework

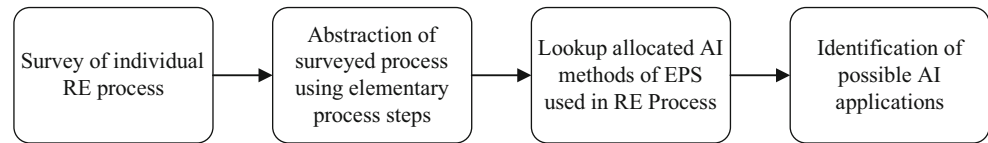
This chapter describes the intended application of the EPS framework. Figure 8 shows the workflow for the identification of AI applications in RE processes, using the EPS framework. After surveying the RE process of interest, the EPS are used to achieve a homogenized representation of the process. For each process step, the EPS catalog can be consulted to receive possibly matching AI algorithms. Finally, the allocated AI algorithms, that are listed in the catalog, can be further evaluated for applicability.

When applying the EPS framework to the exemplary RE process 1 shown in Fig. 9, it can be seen, how the EPS help to overcome shortcomings 1 and 2. Shortcoming 1 is

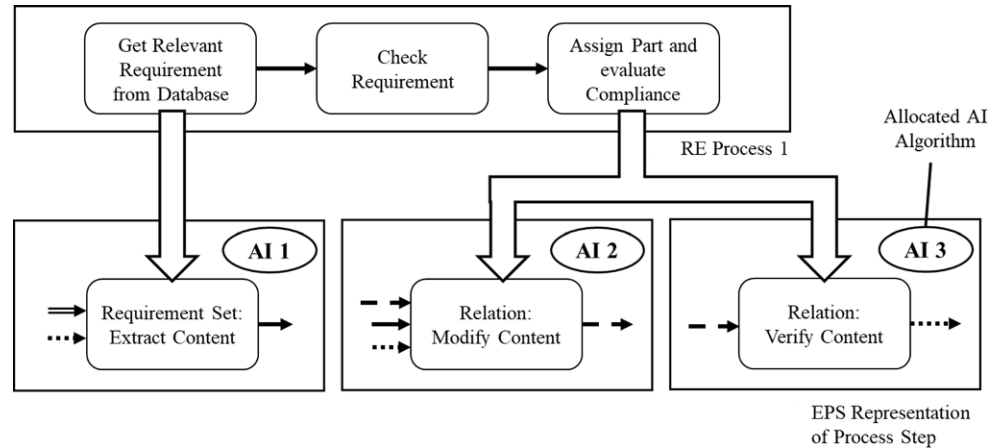
**Fig. 7** Structure of the EPS catalog



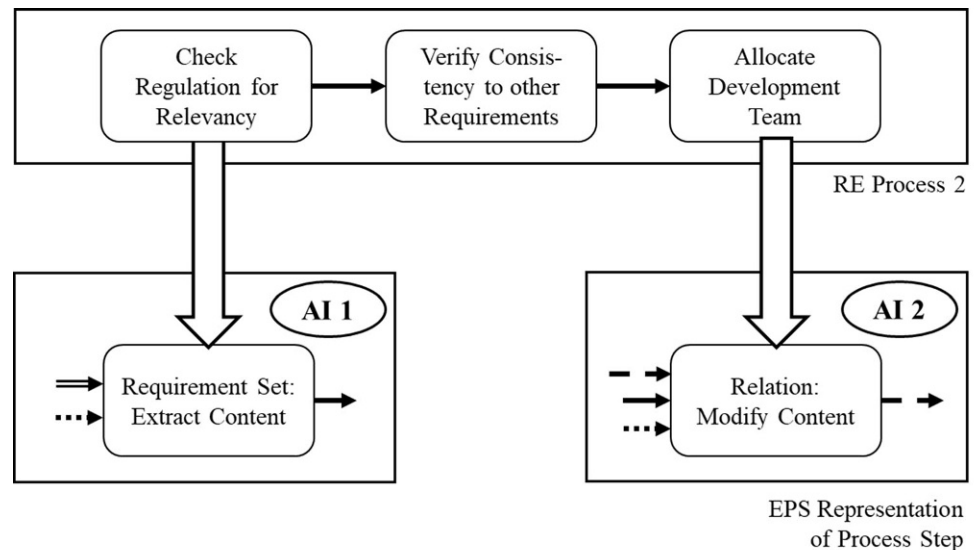
**Fig. 8** Application of the EPS framework to systematically identify AI applications



**Fig. 9** EPS framework applied to the exemplary RE process 1



**Fig. 10** Reusability of the EPS framework: Application to the exemplary RE process 2



addressed by the explicit specification of the in- and outputs of the process step that was formerly only described as *Get Relevant Requirement from Database*. Using the EPS, it becomes clear, that the input is a set of requirements and an information flow, e.g. containing information about the criteria that decide about the relevancy.

In relation to shortcoming 2, the process step formerly expressed through *Assign Part and evaluate Compliance* is split up into two EPS in order to match the scope of the available algorithms. The first EPS uses the RE flow type *Relation* as the main flow and *Requirement* and *Information* as secondary flows. This EPS expresses the assignment of a *Part*, conveyed through the *Information* flow, to a *Requirement*. The Output contains the established *Relation*. The

second EPS expresses the verification of the established relation and delivers the result, i.e. *Information*, about the *verification* as an output.

Figure 10 illustrates the reusability aspect of the EPS framework. The EPS framework is now applied to the formerly introduced exemplary RE process 2. It is revealed that two of the process steps can be represented with the same EPS that are used in RE process 1. Because the AI algorithms *AI 1* and *AI 2* have already been allocated to the EPS, the possible AI applications can be easily identified and further assessed. This addresses the last of the recognized shortcomings, that are described in Chap. 3.



## 6 Conclusion & future work

In this paper, an approach for the systematic identification of possible AI applications in RE processes is presented. Chapter 2 shows, that the application of AI algorithms for RE tasks is a vital field of research. While the ideas for supporting RE tasks by AI algorithms are plentiful and even developed successfully, a disproportionately small number of implementations in practice was found. One of the reasons identified, is the lack of a framework, that allows for systematic assessment of actual RE processes as they are carried out in practice. Three shortcomings of current modelling techniques of RE processes are recognized and described. The EPS framework, conceived by using an analogy to the proven systems engineering strategy of [29], addresses these shortcomings, which is shown by an application of the framework to exemplary RE process steps in Chap. 5. The presented EPS are derived based on a combination of the underlying data model of requirements and basic operations that can be performed on the individual data artifacts of requirements. Within the scope of a collaborative research project with industry participation, the authors were able to use the EPS to homogenize highly individual RE processes that were captured by surveying requirements engineers from multiple departments.

Further research has to be conducted to allocate a significant amount of suitable AI algorithms to the set of EPS, in order to make full use of the proposed framework. A substantial basis could be the assignment of AI algorithms to RE phases of [25]. Although a careful literature research into requirements engineering and digital representation of requirements has been carried out, the authors acknowledge that this set of EPS still can be refined or expanded to cover further aspects of RE processes. Rather than proclaiming completeness of the presented EPS, this research highlights the underlying methodology that can accomplish a systematic identification of AI applications in RE.

**Acknowledgements** The authors want to thank the BMWK for funding the project KIZAM (Künstliche Intelligenz im Ziele- und Anforderungs-management) on basis of a decision of the German Bundestag that made the presented research possible.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view

a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Pahl G, Beitz W, Feldhusen J et al (eds) (2007) Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung, 7th edn. Springer-Lehrbuch. Springer, Berlin, Heidelberg
- Ehrlenspiel K, Meerkamm H (2013) Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit, 5th edn. Hanser, München
- Young R (2004) Requirements engineering handbook. Artech House, Norwood
- ISO/IEC/IEEE (2018) ISO/IEC/IEEE International Standard—Systems and software engineering—Life cycle processes—Requirements engineering. ISO/IEC/IEEE 29148:2018(E), pp 1–104 <https://doi.org/10.1109/IEEESTD.2018.8559686>
- Wnuk K, Regnell B, Berenbach B (2011) Scaling up requirements engineering—Exploring the challenges of increasing size and complexity in market-driven software development. In: Berry D, Franch X (eds) Requirements engineering: foundation for software quality, vol 6606. Springer, Berlin, Heidelberg, pp 54–59
- Iqbal T, Elahidoost P, Lucio L (2018) A bird's eye view on requirements engineering and machine learning. In: 2018 25th Asia-Pacific Software Engineering Conference (APSEC). IEEE, pp 11–20
- Martin S, Aurum A, Jeffery R et al (2002) Requirements engineering process models in practice
- Spütz K, Berges J, Jacobs G et al (2022) Classification of simulation models for the model-based design of plastic-metal hybrid joints. *Procedia CIRP* 109:37–42. <https://doi.org/10.1016/j.procir.2022.05.211>
- Zerwas T, Jacobs G, Spütz K et al (2021) Mechanical concept development using principle solution models. *IOP Conf Ser Mater Sci Eng* 1097:12001. <https://doi.org/10.1088/1757-899X/1097/1/012001>
- Jacobs G, Konrad C, Berroth J et al (2022) Function-oriented model-based product development. In: Krause D, Heyden E (eds) Design methodology for future products. Springer, Cham, pp 243–263
- Höpfner G, Jacobs G, Zerwas T et al (2021) Model-based design workflows for cyber-physical systems applied to an electric-mechanical coolant pump. *IOP Conf Ser Mater Sci Eng* 1097:12004. <https://doi.org/10.1088/1757-899X/1097/1/012004>
- Sharma S, Pandey SK (2013) Integrating AI techniques in requirements phase: a literature review. *IJCA Proceedings on 4th International IT Summit Confluence*, pp 21–25.
- Breuninger J, Kücher F, Misic N (2021) Künstliche Intelligenz im Requirements Engineering. *SKILL* 2021:63–74
- Partsch H (ed) (2010) Requirements-Engineering systematisch. eXamen.press. Springer, Berlin, Heidelberg
- Hull E, Jackson K, Dick J (2011) Requirements engineering. Springer, London
- Robertson S, Robertson J (2012) Mastering the requirements process: getting requirements right, 3rd edn. Pearson Education, Sydney
- Kumar B, Pandey D (2022) Study of requirements engineering process model. *IJRAR – International Journal of Research and Analytical Reviews (IJRAR)* 9(2):135–141
- Kotonya G, Sommerville I (1998) Requirements engineering: Processes and techniques. Worldwide series in computer science. Wiley, Chichester
- Macaulay LA (1996) Requirements engineering. Applied computing. Springer, Berlin, Heidelberg

20. Loucopoulos P, Karakostas V (1995) System requirements engineering. McGraw-Hill international series in software engineering. McGraw-Hill, London, New York
21. Valentini U, Weißbach R, Fahney R et al (2013) Requirements Engineering und Projektmanagement. Springer, Berlin, Heidelberg
22. Rupp C (2014) Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil, 6th edn. Hanser, München
23. Selic B, Bock C, Cook S et al (2015) OMG unified modeling language (version 2.5)
24. Barenkamp M (2020) Künstliche Intelligenz in der Softwareentwicklung. Wirtsch Inform Manag 12:120–129. <https://doi.org/10.1365/s35764-020-00235-5>
25. Zhao L, Alhoshan W, Ferrari A et al (2022) Natural language processing for requirements engineering. ACM Comput Surv 54:1–41. <https://doi.org/10.1145/3444689>
26. Briand LC (ed) (2007) Future of software engineering. IEEE Computer Society, Los Alamitos
27. Knuth DE (2011) The art of computer programming. World student series edition. Addison-Wesley, Reading, Munich
28. Pohl K, Wohed P, Doemges R et al (1995) Applying AI techniques to requirements engineering: the NATURE prototype
29. Koller R, Kastrup N (1998) Prinziplösungen zur Konstruktion technischer Produkte, 2nd edn. Springer eBook Collection Computer Science and Engineering. Springer, Berlin, Heidelberg
30. OMG (2016) Requirements Interchange Format (ReqIF) version 1.2
31. Hood C, Wiedemann S, Fichtinger S et al (2008) Requirements management. Springer, Berlin, Heidelberg