A hybrid knowledge graph-based XAI approach to process analysis with an explanation user interface

Anne Füßl [a,*] and Volker Nissen [a]
[a] *Institute for Business and Information Systems Engineering, Technische Universität Ilmenau, Germany*
E-mails: anne.fuessl@tu-ilmenau.de, volker.nissen@tu-ilmenau.de

Abstract. In the field of consulting, the effective use of artificial intelligence (AI) depends on the ability of both the consultant and the client to understand the results generated by the technology. Our knowledge graph-based approach to explainable process analyses represents a hybrid AI approach that integrates symbolic approaches to structured knowledge with interactive machine learning methods. Algorithmic procedures are traceable and analysis results are presented in a human-readable form. In order to facilitate the presentation of identified weaknesses and suitable improvement measures of analyzed business processes in a manner that allows for intuitive comprehension and enables human-in-the-loop interactions, it is essential to develop an explainable, user-friendly interface. While considerable attention has been devoted to the computational aspects of generating explanations, there is a clear need for further research into the design of user interfaces for explanations (XUI). To this end, a systematic literature review was conducted, and 29 identified explanation components were summarized in a design catalog. The implementation of a prototype XUI for our KBXAI-PA approach serves to demonstrate and evaluate selected design components from the perspective of process analysts.

Keywords: Explanation user interface, hybrid XAI, knowledge graph, process analysis, consulting self-service

1. Introduction and objective

In practical applications of artificial intelligence (AI) systems, a notable challenge is the lack of transparency and traceability in AI results, which can contribute to a general sense of distrust and reluctance to rely on these systems. This phenomenon, termed *algorithm aversion* in the literature, refers to the tendency to prefer human-made decisions over those generated by algorithms [6]. The use of AI applications is associated with substantial modifications to operational working methods and processes. With regard to process acceptance research [30], it is assumed that processes that provide process participants with feedback or explanations on the process status during execution are more likely to be accepted than processes that are less transparent for process participants. The implementation of Explainable Artificial Intelligence (XAI) systems that provide understandable explanations of AI results and incorporate human interactions through the use of a user-friendly interface has the potential to mitigate the occurrence of algorithm aversion and enhance the acceptance of AI modified processes.

The objective of XAI is to develop explanation models for generating results and decisions that can be interpreted by human users through a form of explanation. Consequently, an explanation model discloses the algorithmic decision pathways within an XAI system [2]. In order to facilitate the presentation of results from explanation models

*Corresponding author. E-mail: anne.fuessl@tu-ilmenau.de.

in a manner that is readable, comprehensible and understandable for users, XAI approaches also necessitate the incorporation of an eXplanation User Interface (XUI) [19]. XUIs offer users a comprehensive overview of all outputs in the form of explanations. They facilitate a simplified description of generated results that can be interpreted by humans and enable interactions between the system and the user [5].

The majority of XAI research is concerned with the computational aspects of generating explanatory models. However, there is still a lack of research on the human-centered design of XUI [5, 34]. Nevertheless, it is imperative that users possess a fundamental comprehension of the manner in which results are generated, as this is a prerequisite for their optimal utilization and adaptation. XUIs also facilitate the acquisition of high-quality human feedback, which can then be employed for the purpose of learning algorithms [17]. The design of XUIs represents an appropriate avenue for providing explainable and usable AI, as well as contributing to cognitive support in human-AI interaction [37].

Consulting services constitute a domain of personal services, characterized in the traditional sense by a close and frequent interaction between consultants and clients. The application of AI technologies in the consulting industry has exhibited a gradual increase in recent years, driven in part by the global pandemic caused by the SARS-CoV-2 virus [31]. Complex consulting technologies, such as process mining for analyzing business processes based on event log data, or analytical self-service tools, necessitates a combination of technical and analytical expertise to facilitate the interpretation of results in a manner that is aligned with customer expectations. XUIs can facilitate the presentation of AI results in a intelligible manner to both consultants and their clients, and enhance interactive learning models through the verification of results based on user feedback.

Our hybrid Knowledge graph-Based eXplainable AI approach to Process Analysis (KBXAI-PA) enables the generation of comprehensible results concerning deficiencies and the identification of suitable improvement measures, based on the application of deduction algorithms and traversed result paths [12, 15]. In previous design cycles, an explanation component was created with the objective of providing traceability of process analysis results [12]. Additionally, an interactive machine learning model was integrated with the intention of incorporating user feedback [13]. For a comprehensible result presentation that makes it straightforward for those without process analysis expertise to understand the reasons behind the identified weaknesses and the suitability of the proposed improvement measures, the development of an understandable and intuitive XUI with user interaction is currently in progress. In this context, a component catalog for the design of XUIs was created, which was partially demonstrated and evaluated on the basis of a prototypically implemented XUI [14].

The present article outlines the implementation and evaluation of additional design components for an XUI of our KBXAI-PA approach and builds on an earlier publication on this topic [14]. The research method follows the Design Science Research (DSR) process according to Peffers et al. [33]. First of all, we introduce the basics of human interactions and user interfaces in the context of interactive learning methods. Section 3 presents a systematic literature review according to Webster and Watson [36] for the identification of existing XUIs in the literature, their analysis and the derivation of suitable design components in a design catalog. In Section 4, we introduce the current state of our hybrid KBXAI-PA approach, including the architecture, process analysis procedure, reconstructions of result paths as well as the interactive learning method. Afterwards, Section 5 demonstrates an extended development of the XUI for KBXAI-PA by using selected design components. Finally, in Section 6, we present a detailed evaluation of the implemented XUI through several expert interviews. Section 7 provides a synthesis of the principal findings, a critical review thereof, and an indication of prospective avenues for further research.

2. Basics of human interactions for XAI

Explainable Artificial Intelligence (XAI) represents a research area that emerged with the increase in AI technologies and the consequent need for their applicability. XAI focuses on the development of methods and models to generate explanations of AI results in a way that users can understand, thus reducing the tension between AI performance and explainability [19, 28].

With the point in time at which explanations are generated, three generic explanatory approaches can be distinguished [3, 28]. *Ante-hoc* approaches include exploratory, mathematical-statistical analyses of data sets that serve as input for AI systems to capture data-describing metrics and recognize implicit patterns (e.g., through multivariate methods) in the underlying data. *XAI by design* approaches aim to develop models that are inherently transparent, thereby enabling better explainability of system behavior through small, manageable modules. Finally, *post-hoc* explanatory approaches can be applied to black-box models using model debugging, for example to apply model-agnostic procedures to trained models and provide explanations for the model's decision-making [3, 28].

In this context, a tension emerges between completeness and interpretability. Completeness necessitates the integration of all information deemed necessary and relevant for explainability and decision-making. According to [3] and [4], the interpretability of learned models can be divided into three succeeding levels: *syntactic* (whether a model is readable), *epistemic* (whether a model can be related to background knowledge), and *pragmatic* interpretability (whether a model meets the information needs of users and is plausible). The explanatory power of an XAI approach is contingent on the completeness of all input and output data, as well as information regarding system behavior and the interpretability of AI results. Transparent white-box models generally possess a higher degree of explanatory power than black-box models.

However, not even so called *XAI by design* approaches, which might possess explicit formal semantics, manifest inherent interpretability. There persists a considerable challenge in transitioning from symbolic artifacts (e.g., rule sets or decision trees) to explanations that align with ontological commitments of information structures and processes, while concurrently generating systematically meaningful explanations [18].

During the application of AI systems, users form mental models of systems, objects and processes through their human perception, on the basis of which they try to explain and predict the functioning of AI systems. The improvement of users' mental models can contribute to increasing the explanatory power by developing and using suitable eXplanation User Interfaces (XUI) [10]. The utility of XUIs depends on their design, what should be explained (content of an explanation) and in which form (presentation form of an explanation)[20]. Originally, XUIs were used for decision support in expert systems, in recommender systems as well as in Interactive Machine Learning (IML) [10].

IML forms an intersection between the design of an XUI and an XAI system [17]. The user is involved in the training process by using human input in the selection, creation and labeling of instances [9]. IML was introduced to the Human Computer Interaction (HCI) community in 2003 by Fails and Olsen [11]. Compared to classic ML, IML is characterized by faster (model adaptation at the time of user feedback), more targeted (adaptation of specific aspects of the model) and incremental (small adaptations without major model changes) model adaptations [1]. This allows users to interactively examine the effects of their actions and adjust subsequent inputs to achieve a desired behavior.

Chromik and Butz [5] supplement the description of an explanation support according to Moore and Paris [27] with interaction strategies and design suggestions for interactive XUI and consolidated them into four design principles. Design principle 1, *complementary naturalness*, aims to supplement visual explanations (which illustrate the internal functioning of an AI system) with natural language explanations. The combination of visual cues and textual explanations can promote user understanding, user interaction and the adaptation of learning models.

Responsiveness through progressive disclosure comprises design principle 2. Studies have shown that there is a fine line between 'no explanation' and 'too much explanation'. The user's individual need for explanation influences this boundary [26]. The second design principle represents an iterative approach in which rather general information is provided to the user and detailed information is complemented by requests made by the user. Answering why-questions, for instance, is an essential requirement for comprehensibility [24]. However, it should be noted that why-explanations can also be highly informative. Consequently, it is recommended that why-explanations be structured in a manner that aligns with the individual level of knowledge of the user, thereby ensuring that the complexity and depth of the explanation are commensurate with the user's cognitive capacity. Moreover, the efficacy of why-explanations can be further enhanced by the incorporation of example-based explanations, which have been shown to enhance the comprehensibility of explanations by providing multiple examples that illustrate the same behavior.

Design principle 3 includes *flexibility through multiple ways to explain*. People acquire knowledge in different ways. Páez [32] distinguishes between two ways in which people understand: understanding the cause (based on real observations and examples) and objective understanding (based on simplified ideal models). The combination of different explanation contents (e.g. local and global explanations, counterfactual explanations, example-based explanations) and forms (e.g. tabular, diagrams, textual or colour highlighting) can increase the individual gain in knowledge by complementing several explanation methods. Local explanations are specific to a particular case, while global explanations offer explanations that are generally valid. For example, a global explanation would be that Granny Smith apples are, in general, green, with yellow dots in some cases. A local explanation, on the other hand, would be that a particular Granny Smith apple is green with small yellow dots and red in some places. Counterfactual explanations provide a hypothetical framework that delineates the potential outcomes resulting from alternative decisions or varying initial circumstances [21]. The relevance of features that have led to a result and on which an explanation is based can be represented by visual forms of explanation [23].

The fourth design principle, *sensitivity to the mind and context*, aims to provide functions for adapting explanations in the XUI to the mental model and context of the recipient of the explanation. Users react differently to different forms of explanations depending on existing biases. For this reason, it is necessary to offer personalised and adaptable explanations [7].

The design principles provide a general overview of design objectives to be considered when developing XUIs. However, it is not suggested how these can be implemented through specific design components in a user interface (e.g. through textual, visual or tabular views, filter and sorting functions or a dialogue). The following section examines existing XUIs with the aim of determining suitable design components for an XUI by considering the design principles of Chromik and Butz [5].

3. Design catalog for explanation user interfaces (XUI)

3.1. Overview of literature research

XUIs using IML methods have a variety of design components for presenting results and adapting underlying learning models that need to be selected, described and implemented as part of XUI development. A literature review of existing XUIs, starting with publications in 2003, the year in which Fails and Oslen [11] first mentioned the linking of IML methods in the context of HCI, served to identify design components, which were subsequently structured in a design catalog with categorizing the components based on a concept matrix according to Webster and Watson [36]. Accordingly, the design catalog is a tool for selecting appropriate design components and specifying their requirements for an XUI under development.

In conducting the literature search, the following search query was employed: "Design" AND "explanation interface" AND "interactive" OR "interactive machine learning" AND "user interface" AND "explainable". The databases presented in Table 1 were selected for the literature analysis on the basis of their scientific focus. The overall results based on the search string were initially evaluated for suitability in terms of title and abstract, after which duplicates were excluded. Subsequently, the papers were examined for identifiable design components using full-text analysis, which resulted in 28 articles being identified. Further forward and backward searches yielded seven additional papers.

Table 1
Number of literature results

Database	Overall result	Suitable
ACM Digital Library	68	4
Google Scholar	95	7
IEEE Xplore Digital Library	49	3
Science Direct	70	5
Springer Link	139	2
Total number of suitable literature		21 + 7

As a result of the literature review, 28 interactive XUIs were identified and categorized in a concept matrix. A total of 22 are utilized for data, text, or content analysis, of which eight include a Question-Answering-Dialog (QAD). Six XUIs are employed for image analysis, one of which incorporates a QAD, and one XUI is dedicated to video analysis. The identified XUIs are based on a variety of XAI methods, including decision trees, generalized additive models, neural networks, graphs, support vector machines, random forests, Shapley values, local interpretable model-agnostic explanations (LIME), and different methods for black-box and white-box models. Moreover, the identified XUIs can be distinguished according to the user group to which they are targeted. AI experts (in five cases) encompasses individuals engaged in the development, analysis, and management of AI systems. These professionals may include software developers, data scientists, and project managers. The user group of domain experts (23 times) comprises all other target groups, including model users and users affected by model decisions who lack expertise in machine learning. A total of 25 of the identified XUIs underwent evaluation through user studies and case studies, while three XUIs were not evaluated. The following section presents the identified design components in the form of a design catalog.

3.2. Design catalog of XUI components

The design catalog comprises 29 design components, which can be classified according to the design principles (DP) as defined by Chromik and Butz [5], as illustrated in Table 2, column four. DP 1 encompasses the integration of visual and textual forms of explanation. DP 2 is oriented towards a user-centric, interactive explanation process. DP 3 combines different explanatory content, while DP 4 includes individual customisations.

Table 2
Design catalog with components and their frequencies [14].

Category	Design component	DP	Frequency (n=28)		selected and evaluated using KBXAI-PA
form of explanation	visual	#1	●	89.3 %	*x*_1
	textual	#1	●	75.0 %	*x*_2
	multiple views	#2	◕	60.7 %	*x*_3
	numerical	#1	◕	46.4 %	
	table	#2	◑	39.3 %	
	dependency diagram	#2	◑	32.1 %	
	chat-based	#1	◔	14.3 %	*x*_4
	why-explanation	#2	◔	14.3 %	
content of explanation	global	#3	●	89.3 %	*x*_5
	local	#3	●	85.7 %	*x*_6
	counterfactual	#3	◕	50.0 %	
	example-based	#3	◔	21.4 %	*x*_7
interaction	feature relevance	#3	◕	60.7 %	*x*_8
	search function	#2	◑	28.6 %	*x*_9
	accuracy indicator	#3	◑	28.6 %	*x*_10
	feature distribution	#3	◑	28.6 %	
	visualization of changes in feature values	#3	◑	28.6 %	
	quick-info	#3	◑	28.6 %	*x*_11
	filters	#2	◑	25.0 %	*x*_9
	sorting	#2	◑	25.0 %	*x*_9
	comparison of multiple instances	#3	◑	25.0 %	
	slider	#4	◑	25.0 %	
	feature selection	#2	◔	21.4 %	
	control elements	#4	◔	21.4 %	*x*_9
	prioritizing	#4	◔	7.1 %	
adaption	instance correction	#4	●	67.9 %	*x*_12
	feature correction	#4	◕	53.6 %	*x*_12
others	font design	#4	◔	14.3 %	
	video	#3	◔	3.6 %	

Upon analysis of the design components, four categories were formed for structuring purposes, as indicated in the first column: Explanation form, explanation content, interaction and adaptation. Two components could not be assigned to any category and are listed under others. The categories represent four essential groups from which selected design components can be used for XUI development.

The frequency of occurrence of the identified components in the literature can be used to assess their relevance. The components are structured as follows: Components with a frequency of over 75% are assigned the value 1: ●, while components between 50% and 75% are weighted with a value of 3/4: ◕. Those elements with a frequency of less than 50% and greater than 25% are assigned a value of 1/2: ◑, while components with a frequency of less than 25% are given a value of 1/4: ◔. Table 2 presents the design components for each category in descending order, according to their frequency of occurrence. It should be noted that the catalog does not claim to be exhaustive.

A review of the literature reveals that visual and textual forms of explanation are employed most often in combination in several views for global and local explanatory content. In contrast, tabular explanations were employed solely 11 times, predominantly for the presentation of numerical data. Over 60% of the identified papers employ a combination of multiple explanation forms. Half of the identified papers employ counterfactual explanations, which utilize what-if scenarios as a means of providing an explanation. Furthermore, adaptations of features (53.6%) or concrete instances (67.9%) are also part of an XUI in the majority of the analyzed articles. Instance adaptations pertain to a specific analysis result, whereas feature adaptations relate to an underlying machine learning model. This illustrates that the user-centered modification of the learning model is a pivotal aspect in the development of an XUI. An examination of the interactions reveals that a number of functions were identified with medium frequency. In the majority of XUIs (60.7%), the relevance of features that led to a result and on which an explanation is based can be retrieved through user interaction. The remaining functions are employed to varying degrees in less than 30% of the reviewed articles, which is likely attributable to the specific domain of application. Search and filtering functions, as well as functions for sorting and feature selection were counted six to eight times for each, although these should be main functionalities. The results of the literature research indicate that the chat-based explanation form and why-explanations, which provide information about the consequences of a recommended result, are used less frequently. Furthermore, the utilization of customized font design to accentuate explanations or explanation videos is also infrequent.

The design catalog offers an overview of selected XUI components and their respective frequencies. For those engaged in the design of XUI systems, it provides a foundation for the selection of appropriate components for the presentation of XAI results in a simplified and interactive manner, in accordance with the design principles for XUI [5]. In a first prototype, an XUI for the knowledge graph-based XAI approach to process analysis (KBXAI-PA) was demonstrated and evaluated [14] based on a part of design components marked in the last column of Table 2. The present paper provides the demonstration of all the labelled components of Table 2 and a more comprehensive overview of the XUI evaluation. In the following section, we initially introduce our KBXAI-PA approach.

4. Current state of KBXAI-PA

4.1. Concept and architecture

The Knowledge graph-Based XAI approach to Process Analysis (KBXAI-PA) facilitates the identification of deficiencies and suitable improvement measures in business processes on the basis of a knowledge graph architecture enriched with expert knowledge. In the context of consulting practice, it is not possible to automatically derive such conclusions from process models without the necessary semantic context. Process mining, an automated method of process analysis, employs log data from IT systems to identify hidden process information within these systems [35]. However, manual tasks and implicit process-related knowledge are not taken into account. Furthermore, the interpretation of process mining results requires the application of specific analytical knowledge to derive practical and useful conclusions.

A review of the literature revealed no freely available knowledge base in the form of an ontology that would be suitable for identifying weaknesses and improvement measures in business process analysis. Noy and McGuinness propose the creation of an ontology from scratch for domain-specific knowledge bases [29]. For a systematic process analysis with a context-specific knowledge base, analogous to the expert knowledge of a process consultant, we employ an ontology-like knowledge graph architecture with deduction algorithms and inductive learning mechanisms [16] to develop a knowledge base for process analysis in the form of a knowledge graph and to implement the process analysis procedure algorithmically.

The architecture of the knowledge graph can be described as a five-layer model, as proposed by [16]. The data input layer (L4) is followed by the data transfer layer (L3), which is in turn succeeded by layers for information processing and knowledge representation of concrete elements (L2) and abstract elements (L1). The solution layer (L0) contains all activated nodes per analysis. With the exception of L0, the aforementioned layers consist of different element classes that fulfil their respective layer functions, as illustrated in Figure 1.

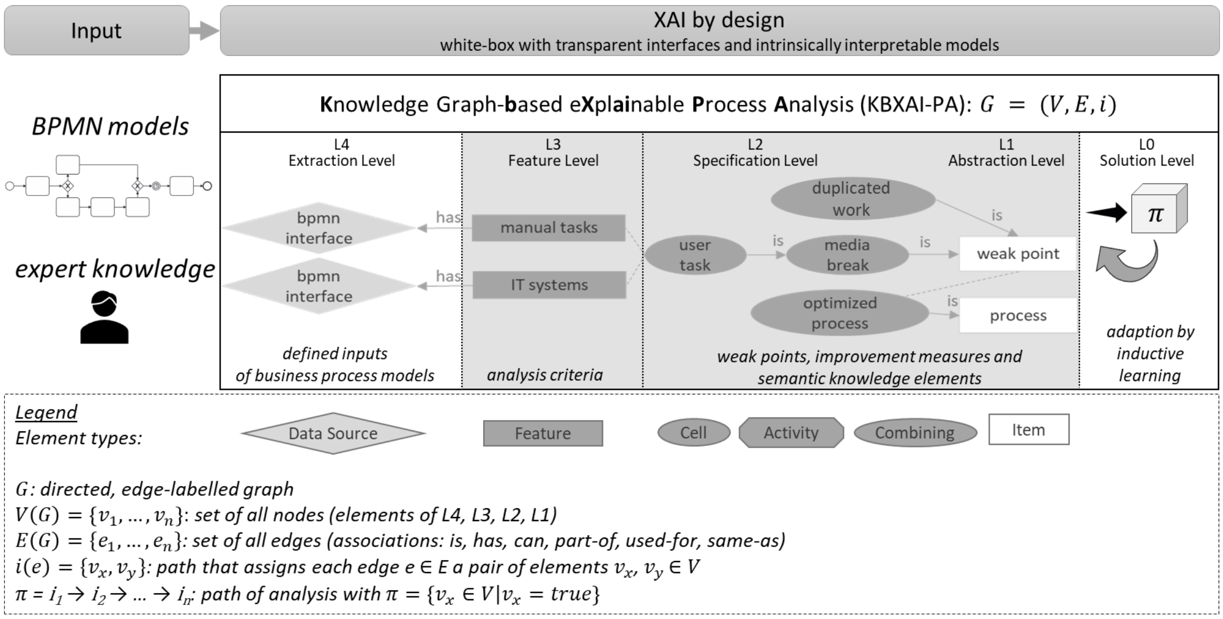

Fig. 1. Architecture of KBXAI-PA according to [14]

Level L4 is employed for the interconnection of external *Data Sources*, which represent the input of an analysis. XML-based BPMN process models are read out via specified interfaces. The data input elements, depicted as a rhombus (see Figure 1), necessitate a minimum of one data transfer element at level L3. The purpose of data transfer elements is to facilitate the transfer of data in the appropriate format (e.g. integer, string, single or multiple choice) from level L4, here for instance from a BPMN process model, to analyze and activate knowledge elements about process deficiencies and improvement measures at level L2. At level L3, the analysis criteria of the process analysis are defined as *Feature* elements. Such criteria may include the number of manual activities or repeated activities, which indicate potential weaknesses in process models. Examples of such weaknesses are media breaks or duplication of work. Weak points and improvement measures are modeled in the information processing levels L2 and L1 (comparable to A- and TBox). At level L2, the element types are defined as follows: *Cell* (concrete element labeled as circle), *Combining* as a specific type of *Cell* with more than one term (also labeled as circle), and *Activity* (concrete element denoting an action, labeled as hexagon). At level L1, the abstraction layer, an *Item* represents an abstract element in the form of a square.

The basic association classes, including *is*, *can*, *part of*, *used for* and *same as* facilitate the establishment of semantic relationships between the elements. The nodes of levels L2 and L1 may be equipped with calculation formulas, which are used to obtain the Boolean values *true (1)* or *false (0)*. A result expression $r(v_x)$ is defined for the conditional activation of an element v_x and is characterized by *Constraints* (dashed edge, see Figure 1). The weak point *user task* as a type of *media break* (see Figure 1) is identified when a process is concurrently engaged in both manual tasks and IT system operations. The associated result expression of the node *user task*: $r(v_{userTask}) = true \leftarrow r(v_{manualTask}) > 0$ && $r(v_{ITsystem}) > 0$ leads to activation *true*, while all *is*-successor nodes are also activated (here: $v_{mediaBreak}$ and $v_{weakPoint}$). The *Constraints* between $v_{weakPoint}$ and $v_{optimizedProcess}$ indicates (dashed edge, see Figure 1) that an optimized process exists as long as no deficiencies have been identified in a process model. The following calculation formula is defined as follows: $r(v_{optimizedProcess}) = true \leftarrow r(v_{weakPoint}) == false$. Accordingly, the node $v_{optimizedProcess}$ represents the initial node of the process analysis procedure, see Section 4.2.

Each process analysis contributes to the refinement and expansion of a reusable knowledge graph through Inductive Logic Programming (ILP), as introduced by [16] in the initial architecture. This expansion involves the incorporation of novel knowledge elements derived from input data of process models. An example of this incorporation is the integration of new IT systems as supporting systems for improvement measures [15]. Additionally, concrete knowledge elements inherit their attributes to abstract knowledge elements at a higher level [16] (e.g. the abstract element *manual process* inherits a *has*-link to a new element of *analog document* that was identified as a concrete element of *manual task*) [15]. Model adaptations by refining the knowledge graph on the basis of implicit knowledge, as well as checking and correcting analysis results with the aid of user feedback, represent an interactive learning approach, which is presented in Section 4.4.

4.2. Process analysis procedure

Business processes are analyzed in four generic steps of analysis, initiated by deduction algorithms. Deduction algorithms are classified into two categories [16]: abstraction algorithms, which include *Is-it* and *Kind-of*, and concretization algorithms, which encompass *Characterize*, *Parts*, *Like* and *Find*. Abstraction algorithms check knowledge elements of the association class *is* for their Boolean values, activate linked *is*-successor elements of an entry node, and serve to identify weak points.

Algorithm 1 Analysis procedure in pseudocode

```
procedure ProcessAnalysis(BPMN)
Q1: call Is-it optimized process
      if result (Q1) == false
Q2:       call Kind-of weak point
          if result (Q2) is not null
              call Characterize
              call Parts
              call Like
              call Find
Q3:           call Find improvement measure
              if result (Q3) is not null
                  sort result elements by w
                  for each result element of (Q3)
                      calculate Ā
Q4:                   call Find improvement tool
                      if result (Q4) is not null
                          sort result elements by w
                          for each result element of (Q4)
                              calculate Ā
                          end for
                  end for
end procedure
```

The initial analysis step (see Q1 in Algorithm 1) is thus designed to ascertain whether an optimized process exists by examining a process model for the presence of at least one deficiency. Due to the negation of the first analysis step, the second analysis step (Q2) identifies all weaknesses of the process model to be analyzed. Meanwhile, concretization algorithms are triggered depending on their respective association class. For identifying properties of deficiencies, nodes that are linked via *has* or *can* associations are analyzed by the *Characterize* algorithm. The identification of part-whole relationships in process models, such as prerequisites necessary for automatic supplier assignment in a purchase order, is achieved by the *Parts* algorithm through the examination of *part-of* associations in the knowledge graph. In order to identify activities such as *checking*, all synonymous activities linked to the element *check* via *same-as* associations are analyzed using the *Like* algorithm. The *Find* algorithm analyzes all *used-for* links in the knowledge graph for specific purposes, e.g. automatic supplier allocation is used for purchase orders. In this manner, semantic descriptions can be generated that pertain to an identified deficiency (such as: *User task enter customer data can cause errors*). Moreover, in the third and fourth steps (see Q3 and Q4 in Algorithm 1) of the analysis procedure, improvement measures and tools for mitigating the identified weaknesses are determined using concretization algorithms. The process analysis procedure is shown above in the form of pseudocode. Here the variables w for the weighting of nodes and A for the assessment of results are recalculated in the last two analysis steps (see Section 4.4 Interactive learning and assigning weightings).

Process analysis procedure using an example process For the purpose of illustration, the subsequent example process (illustrated in Figure 2) demonstrates a standard processing of an incoming travel request in a travel agency. This process is represented by a simplified BPMN model. The model includes various user and manual activities (marked by BPMN user tasks and BPMN manual tasks), such as checking booking availabilities, entering customer data, asking for missing data and providing tickets. In addition, the model contains four intermediate events and two end events.

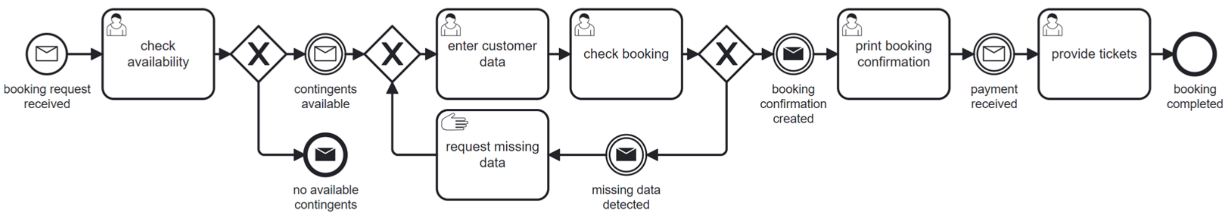

Fig. 2. Sample BPMN model for processing an incoming request in a travel agency

As previously stated in Section 4.2, the node $v_{optimizedProcess}$ of an analysis knowledge graph functions as the initial entry point for process analysis. The initial step in the deduction procedure outlined in question 1 (Q1 of the analysis procedure, see Figure 3) involves examining a process model for the presence of at least one weak point, thereby determining whether an optimized process exists. The presence of at least one deficiency, as indicated by the existence of BPMN user tasks, multiple end events, or missing data, is indicative of the process in question. Consequently, the result value of the entry node $v_{optimizedProcess}$ is determined to be false, thereby answering the first analysis question in the negative.

The second step of analysis (Q2) involves the identification of potential weaknesses in the process model to be analysed, should the initial analysis yield a negative result. With regard to our process model, a substantial number of BPMN user tasks, including printing, checking, and entering, were identified. Furthermore, potential weaknesses were identified, such as information deficits and human decision-making tasks.

This in turn triggers the execution of concretization algorithms, depending on the respective association class. In order to identify all activities that indicate a check, all synonymous activities that are linked to the *check* node in the knowledge graph via a *same-as* association are analysed by calling the *Like* algorithm. A more detailed analysis of potential weaknesses, such as BPMN user tasks with *enter*, can be achieved by using concretization algorithms to determine semantic arguments. To illustrate: The *Characterize* algorithm leads to the identification of the following semantic description: *User task enter customer data can cause errors*.

In line with the findings from question two (Q2), recommendations for improvement measures and tools are formulated in questions three (Q3) and four (Q4) by the concretization algorithm *Find*. In the context of the booking request in Figure 2, the BPMN user task for checking the booking data was identified as a potential weakness as it is currently performed as a human decision task. A human decision task is characterized by a manual or user task that occurs prior to a branching exclusive or inclusive gateway. In order to address the identified deficiencies and the potential enhancements in the knowledge graph, the following measure and improvement tool are proposed for instance: *Revise human decision task can be used to mitigate the possibility of inaccuracies that could arise* *in a human check task. Online booking engine is a tool used for automate data transfer*. Measures are arranged in accordance with their relevance, as determined by user feedback. In instances where user feedback has already been provided for a specific set of weaknesses and improvement measures, the measures with higher ratings are listed first.

In the final two analysis steps (Q3 and Q4), the variables w, which represent the weighting of the nodes, and A, which symbolize the evaluation of the results by users, are recalculated. Result paths are used to evaluate the identified improvement measures with the aid of weightings. The explanation process is illustrated in the following section by presenting the reconstruction of result paths from the process analysis. Subsequently, the interactive learning method and the assignment of weights are introduced.

4.3. *Explanation process with creation of result paths*

The *XAI by design* architecture of KBXAI-PA permits the reconstruction of the paths traversed by the deduction algorithms of the knowledge graph, which serves as the foundation for the explanation process of KBXAI-PA. Figure 3 illustrates the explanation process occurring between the basic architecture and the resulting output.

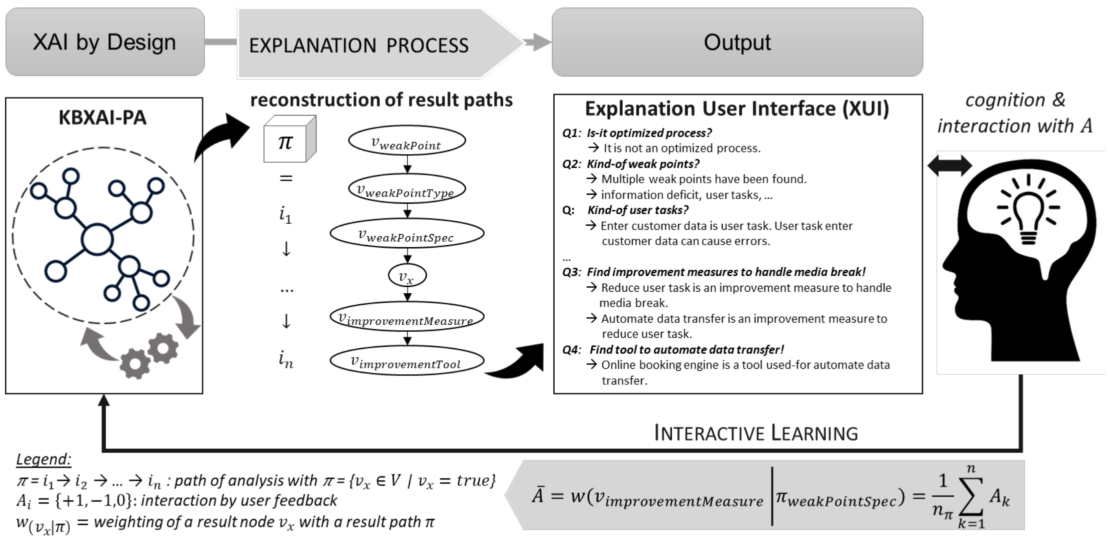

Fig. 3. Explanation process with an exemplary explanation output

The process analysis performed with KBXAI-PA results in the identification of process deficiencies and recommendations for suitable improvement measures. The outcomes are presented via an eXplanation User Interface (XUI), as illustrated in Figure 3 on the right. The XUI not only serves as a means to provide explanations for the results but also operates as a Question-Answering Dialog (QAD).

The objective of a QAD is to enhance the analysis results by collecting user feedback in order to achieve the closest possible alignment between the analysis results and the user's intentions [22]. A natural language dialog is used to facilitate user interaction and enable the verification of identified improvement measures, such as *online booking engine is a tool used-for automate data transfer*, see Figure 3 result of Q4. The semantic links of the nodes, which are based on the association classes that are invoked during the analysis procedure, are used to enrich analysis results with natural language explanations, making it necessary to reconstruct all result paths in the context of a process analysis.

During process analysis, the nodes contained within the knowledge graph are mapped into interpretable models based on decision trees in accordance with the specified algorithmic sequence [12]. An algorithmic sequence consists of various inferences. Each inference, denoted by Q in the analysis procedure or in the XUI (see Figure 3), yields one or more results representing a target node. The result paths of target nodes are reconstructed according to the order in which they are activated. Nodes with the result value *false* are excluded from the result path, meaning that a result path of a target node consists of a series of activated nodes and the corresponding edges. Nodes for which the value has already been calculated in a prior inference are not required to undergo recalculation. Algorithm 2 illustrates the procedure for the path reconstruction of an inference in pseudocode.

Algorithm 2 Reconstruction of a result path regarding an inference

```
procedure PathInferenceResults(inferenceResults)
    filteredResults ← FilterResults(inferenceResults)
    for each resultPath in filteredResults do
        targetNode ← resultPath.target
        UpdateMissingResults(targetNode)
        if not ExistsInResultTable(targetNode) then
            Calculate(targetNode)
            CreateNewEntry(targetNode)
        if resultPath.hasResult then
            entry ← FindResultTableEntry(targetNode)
            StoreResult(entry, resultPath.result)
    end for
end procedure

function FilterResults(inferenceResults)
    validResults ← []
    for each result in inferenceResults do
        if result.value ≠ false then
            validResults.add(result)
    return validResults
end function
```

The target nodes of the analysis procedure (Q1-Q4) enable a result path π to be broken down into generic aggregations (see Figure 3, reconstruction of result paths). A sequence is initiated with a series of weak point types, leading to specific deficiencies. It continues with semantic description elements of a weak point specification, progressing to improvement measures and, if necessary, improvement tools. The interpretable models utilize the same aggregations [12], but are not the primary focus of this article.

As illustrated by the right-hand side of Figure 3, the result output for the sequences Q1 to Q4 of the analysis procedure is presented in abstract form in an XUI. The initial analysis step, designated as Q1, thereby serves as a starting point for the investigation of a process model. In the analysis step Q2, all identifiable weak points of a process model are consequently determined. Furthermore, depending on the associated and activated nodes, additional concretization algorithms are invoked to identify semantic relationships, which then serve as the basis for determining suitable improvement measures in steps Q3 and Q4. The identification of weak points, such as user tasks (e.g. entering customer data), results in the recommendation of improvement measures (in this case, an online booking engine).

A section of the knowledge graph of the KBXAI-PA architecture employed for the analysis of the given process example (see Figure 2) is illustrated in Figure 4. The nodes highlighted in grey, grey-black diagonal hatched and black, represent components of different result paths that are generated by deduction algorithms during the process analysis (π_{1_n}, π_{2_n}, $\pi_{1,2}$). Here, the black nodes represent both activated nodes of a result path π_{1_n} as well as a result path π_{2_n}.

During the course of the analysis procedure (Q1-Q4, see algorithm sequence 1), the result paths will be structured into the generic aggregations, as illustrated in Figure 4 above. It should be noted that the arrow directions in the knowledge graph represent the semantic context, rather than the algorithmic direction of analysis. The initial stage of the analysis procedure involves the examination of all nodes that represent a weak point type, starting at the abstraction level. For instance, the element *media break* can be identified as $v_{weakPointType}$. Subsequently, the elements of the specification level are analyzed, which represent concrete weaknesses as weak point specifications (e.g. *existing user tasks* as $v_{weakPointSpec}$). In addition, all nodes that exhibit semantic relations due to their associations also become part of a result path (e.g. *enter customer data*, *booking request* as v_x).

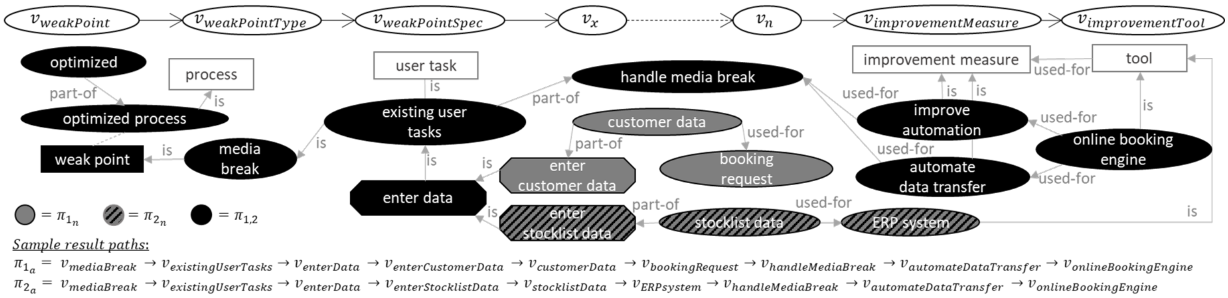

Fig. 4. Result paths π with activated nodes of process analyses [13]

A result path π_{1_a} in which the user task *enter customer data* used for a *booking request* was identified as a weak point specification of the weak point type *media break* and the improvement measure *automate data transfer* using an *online booking engine* as an improvement tool was derived, is as follows: $\pi_{1_a} = v_{mediaBreak} \rightarrow v_{existingUserTasks} \rightarrow v_{enterData} \rightarrow v_{enterCustomerData} \rightarrow v_{CustomerData} \rightarrow v_{bookingRequest} \rightarrow v_{handleMediaBreak} \rightarrow v_{automateDataTransfer} \rightarrow v_{onlineBookingEngine}$. In this way, generated analysis results can be traced in the knowledge graph through their inferences and activated nodes. Their association classes (*is*, *has*, *can*, *part of*, *used for* and *same-as*) are used to generate textual outputs (e.g. *Online booking engine is a tool used for automate data transfer.*), which enable explanation sentences regarding a target node [12].

However, it is not the case that all identified improvement measures are equally relevant for each process. The suitability of improvement measures in relation to a deficiency can be rated higher or lower by assessing the results of process analysis through the lens of user interactions.

4.4. Interactive learning and assigning weightings

Through the incorporation of user feedback via an interactive learning loop of process analysis (see Figure 1), human users are enabled to validate, assess the relevance, and refine the results obtained from the process analysis steps Q3 and Q4 [13]. The verification of process analysis results and the assignment of ratings are contingent on the generated result path, as not every identified improvement measure is equally relevant for an analyzed process. As an illustration, a media break may be identified at the point of entering customer data for a booking request (π_1, see Figure 4), as well as at the point of manually entering a stock list in an ERP system (see π_2). In both cases, automated data transfer appears to be a suitable improvement measure at first glance. However, if customer data also includes travel preferences, which represent a customer-specific input, and a customer is undergoing on-site consultation, other improvement measures appear to be more appropriate.

In order to manage such cases effectively, it is essential to ascertain the generation of results within the context of the various learning phases. In the early development phase of the learning model, the primary objective of the results review is to validate the accuracy of the identified weaknesses and improvement suggestions in relation to the analyzed process. This process is primarily executed by an ML expert with proficiency in both graph architecture and process analysis. To achieve this, semantic links and calculation formulae in the knowledge graph are adapted. The subsequent training phase of the learning model is conducted in real-time, whereby end users (both clients and consultants) assess the relevance of identified weaknesses and proposed improvement measures. These assessments can be conducted on a company-wide basis or tailored to specific clients.

Regardless of the stage of the training phase, for each result path of a weakness, the user has the option of confirming $(+1)$ or rejecting (-1) an improvement measure. In the event that the user has no prior experience with the proposed improvement measure and takes no action, the result of the corresponding result path is given the value null. The weighting w of a result node $v_{improvementMeasure/Tool}$ learned through interaction can be calculated via the mean value of all evaluations $\overline{A}$ for a result path of a weak point specification (see calculation formula in Figure 3).

For demonstration purposes, an erroneous association between a weak point and an improvement measure exists within the knowledge graph, as illustrated in Figure 4. For the identified deficiency of the result path π_{2_a}, the node $v_{onlineBookingEngine}$ is incorrectly proposed as an improvement measure regarding the weak point of *enter stocklist* *data*. If the value of the weighting of a result node assumes the value -1 after more than three evaluation cycles, the calculation formula stored in the result node is adjusted by ILP, which must be checked for correctness during the training phase (the number of evaluation cycles is utilized during the initial training phase and can be altered depending on the necessity for correction). Thereby, the system prompts the user to check and correct, if necessary, the automatically adapted calculation formula of the result node. In this case, the calculation formula $r(v_x)$ of the node $v_{onlineBookingEngine}$ is specified in a domain-oriented manner by expanding it by adding customer data in relation to a booking request. The adapted calculation formula is therefore as follows (additions in bold): $r(v_{onlineBookingEngine}) = true \leftarrow (r(v_{enterData})\ \&\&\ \boldsymbol{r(v_{customerData})}\ \&\&\ \boldsymbol{r(v_{bookingRequest}) == true}$. During the training phase, automatically adjusted result expressions of nodes that represent improvement measures in relation to identified weaknesses are checked and corrected before the process analysis tool is deployed in a live environment. After the initial training phase, this manual checking procedure is no longer necessary.

The integration of user feedback facilitates the refinement of process analysis procedures through each iteration, enabling the prioritization of improvement measures for each identified weakness within a specific domain. This, in turn, enhances the efficacy of deduction algorithms in identifying appropriate improvement measures within the knowledge graph. However, a correct assessment of analysis results also requires a full understanding of the process analysis and how results were generated. In the following, we show the design of an XUI with interaction functionalities in a case study and evaluate the XUI in terms of comprehensibility and traceability of process analysis results.

5. Demonstration of an XUI for KBXAI-PA

The objective of designing an XUI for KBXAI-PA is to present the analysis process and its results in a manner that is easily readable, comprehensible and understandable, as proposed in the XAI goals [19]. The explanation should be understandable to the extent that the user is able to identify indicators (e.g. concrete user tasks) that justify the identification of a weakness (e.g. media break) by considering the path of result generation. Any proposed improvements (e.g. enhancements to automation) should be supported by a detailed argumentation based on the identified deficiencies. The user should be able to interact with the system in order to query detailed descriptions and carry out verifications and corrections to a process analysis, which will then be considered in subsequent analyses.

A number of design components were selected based on a frequency of 1/2 to 1 in relation to the analysis procedure in order to develop an initial design for an XUI for KBXAI-PA, ensuring that each design principle is represented by at least one component (see Table 2, marked in the last column). In the first category, visual (x_1: e.g., excerpts from the result paths in the form of decision trees) and textual (x_2: explanations in natural language) forms, as well as multiple views (x_3: differentiation between end user and analysis expert) and a chat-based form of explanation (x_4: chat-based interactions and generated answers in the form of explanations) are implemented. In order to illustrate different content types of explanations in the demonstration, three types were selected for representation: global explanations (x_5), local explanations (x_6), and example-based explanations (x_7). In the category of interactions, feature relevance (x_8: prioritized order in which the results are displayed), an accuracy indicator (x_10: regarding the recommendation of improvement measures) and quick-infos (x_11: by hovering over individual terms or results) are implemented. In addition, usual functions and control elements (x_9) such as search, filter, and sort are integrated. The functionality of correction (x_12) enables the adjustment of instances or features. In comparison to the preliminary XUI development [14], the design components x_4, x_7, and x_9 were likewise included in the implementation and evaluation process. However, further investigation of all other components is recommended in future research of KBXAI-PA as well as for other XAI approaches. The subsequent exposition will present the selected and implemented design components in the XUI through a demonstrative process analysis.

5.1. Execution of process analyses

The developed XUI has an interactive design and comprises two functional components (x_3): a backend with an administration view for analysis and system experts, and a frontend with a process analysis view for domain experts and end users (see Figure 5). The administration view is utilized for the configuration of analytical procedures,

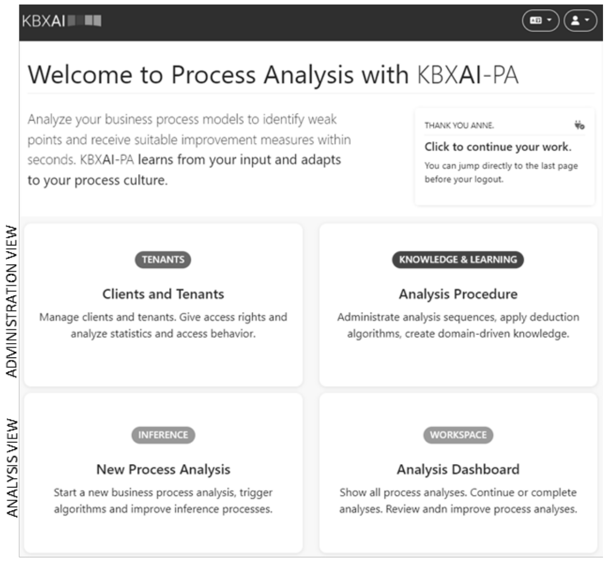

Fig. 5. Starting page of KBXAI-PA prototype

encompassing deduction algorithms and the construction of knowledge graphs. Here, the Clients and Tenants module enables tenant-specific analysis graphs, access rights management, and analysis statistics. The process analysis view provides functions for executing new process analyses and a dashboard of generated analysis results regarding associated tenants.

Initiating a new process analysis necessitates the upload of the process to be analyzed in BPMN format. Additionally, BPMN models generated from previously analyzed event log data by process mining applications can be uploaded via an API. In this manner, a process mining analysis may be refined through the utilization of our knowledge graph-based approach to explainable process analysis.

For illustration purposes, a sample order-to-cash process model with carrier selection and shipping is considered, comprising a number of user tasks, repeatable tasks and media, as well as organizational breaks (see Figure 6). Following the upload of the sample process model, the analysis is executed.

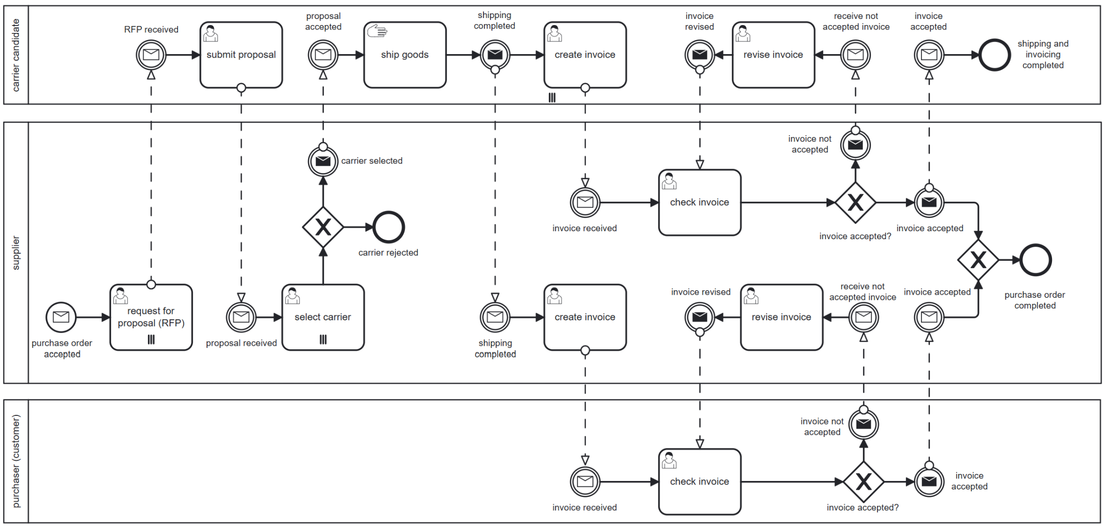

Fig. 6. Sample order-to-cash BPMN model including carrier selection and shipping

A part of the underlying knowledge graph is shown in Figure 7, which is available in the administration view. When performing the process analysis, the nodes traversed by the analysis procedure in the knowledge graph are first checked for weaknesses (e.g. media breaks). Linked predecessor nodes representing weakness specifications (e.g. user tasks such as *scan* or *print*) are run through the deduction algorithms and checked against defined formulas to calculate whether the weakness specification can be identified in the process model (e.g. by identifying and counting message flows between organizational units and IT systems in BPMNs). Data input elements of the graph architecture level L4 (see rhombus in Figure 7) are linked to each node containing a calculation formula with required input values from a BPMN. The labels of the dashed edges, which are also known as *Constraints*, serve to denote stored calculation formula. The number that follows the symbol # in these labels corresponds to the identifier of the specific node. The representation of knowledge graph sections in the administration view constitutes a visual form of explanation (x_1).

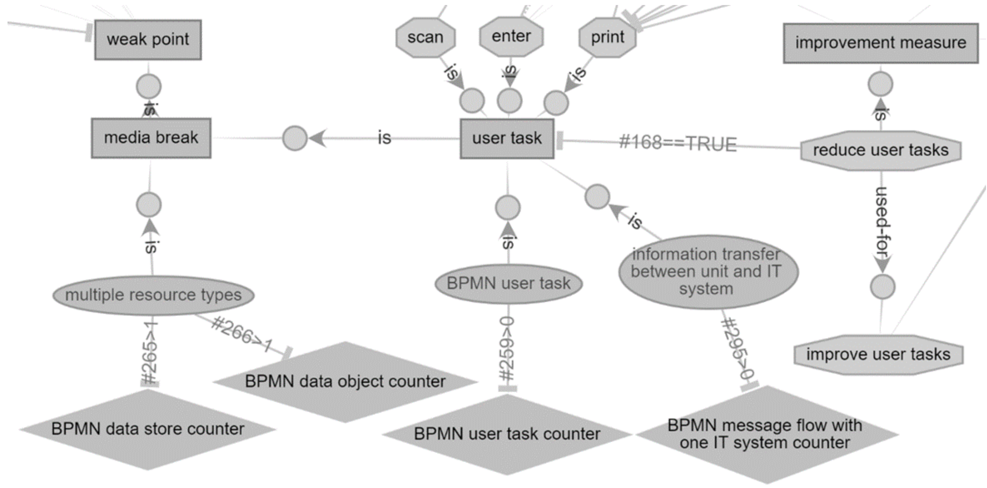

Fig. 7. Section of the knowledge graph

The deduction algorithms examine the linked predecessor nodes for their result values *true* or *false*. In the example of the L4 element *BPMN message flow with one IT system counter* (see right-hand rhombus in Figure 7), an interface checks whether message flows exist between a lane or a pool with at least one IT system in the process model. The corresponding formula is defined as follows $r(v_{infTransferUnitITsystem}) = true \leftarrow r(v_{BPMNmessageFlowITsystem}) > 0$. The node *information transfer between unit and IT system* becomes *true* when the condition of the formula is fulfilled. The result values are retroactively assigned to the successor nodes passed through on the analysis path. In this case, the successor nodes *user task*, *media break* and *weak point* would also receive the value *true*. All other weaknesses modeled in the analysis graph are thus analyzed and lead to the identification of appropriate improvement measures [12].

5.2. Procedure results of process analyses

The view of the analysis results represents the XUI at its core, see Figure 8. It comprises an *Analysis overview* containing metadata related to a respective process analysis, such as a timestamp, the used analysis graph, an overall result, procedure results on process deficiencies, and finally recommended improvement measures. A search function has been incorporated into the header, and sorting and filtering functions are available in the sections of weak points and improvement measures (see design components x_9). The overall result of the sample order-to-cash process indicates that the analyzed process model exhibits potential for optimization.

The area *Procedure results: weak points* displays all deficiencies analyzed in a process model, including a description and an explanation. In Figure 8, an extract of the identified weaknesses is represented (positions 1-5). Each

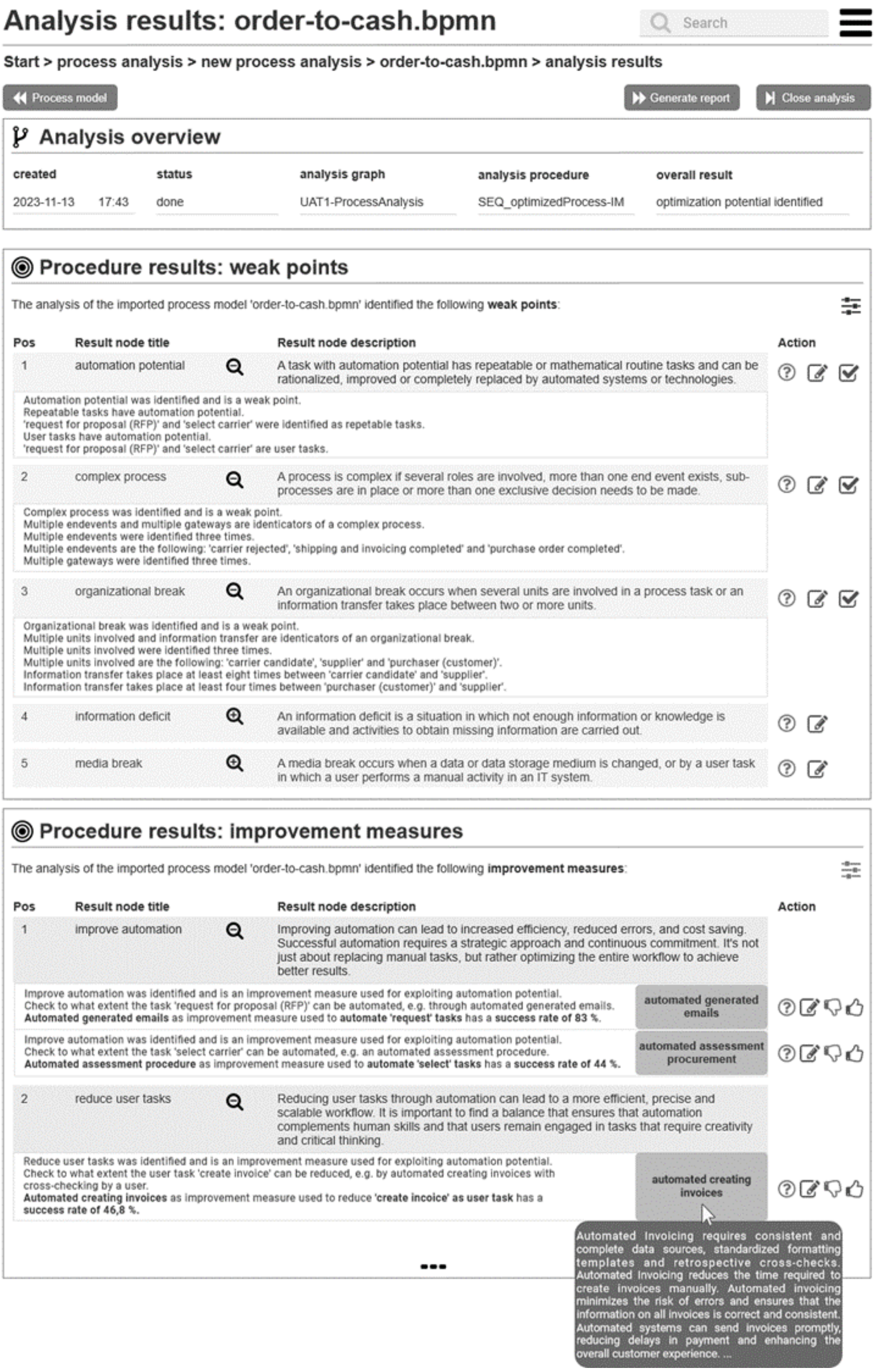

Fig. 8. Process analysis view: Procedure results (extended from [14])

listed weak point has a result node title and a description of the result node. The zoom magnifier icon is used to display explanations of how process deficiencies were identified in the knowledge graph. There are two sentences with global explanations for the first identified result *automation potential* (x_5). The third sentence shows which tasks in the process model have automation potential and for what reason (because of *repeatable tasks*) (x_6). The explanation phrases are created according to the traversed result paths of a process analysis and taking into account activated nodes and their association classes.

A log of all result path runs π_n, including the calculated results and activated nodes with the value *true*, is documented in the administration view (refer to x_3). An extract of logs for identifying the improvement measure *automated generated emails* in relation to a request task (here: *request for proposal (RFP)*) as a repeatable task with automation potential is shown in Figure 9. It shows the algorithm that generated the result path (here: *Find*), a timestamp with an ID, the respective source and target nodes along a path, as well as the results of the target nodes. It is evident that the result values of the traversed target nodes are assigned retroactively after determination (here: the initial five lines exhibit no results, only subsequent to the determination of the identification of the task *request*, as depicted in the accompanying figure. All node pairs of a result path with the result value of *true* are transformed into natural language statements (x_2), which are displayed as explanation content in the process analysis view for domain experts and all other end users (see Figure 8).

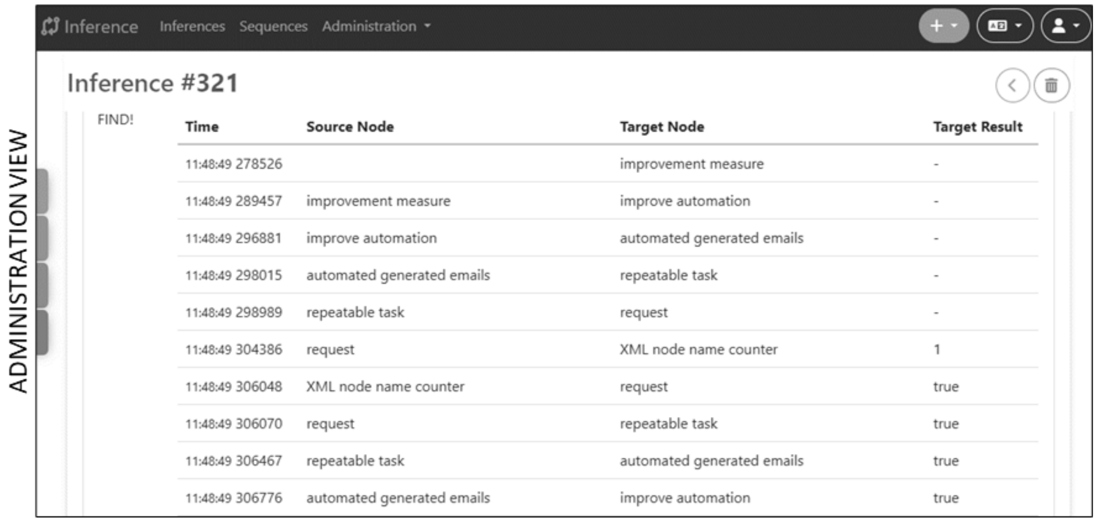

Fig. 9. Administration view: logs of result paths

The *Procedure results: improvement measures* are given for each weakness, as shown in Figure 8 as an example for *automation potential*. In addition to the descriptions of the result nodes on the right-hand side, two specific improvement measures with success rates are suggested by clicking on the magnifying glass icon. The success rate reflects the result weight w, which is calculated by user feedback in the interactive learning process [13]. Thus, the success rate is higher if the proposed improvement measures for a weakness are more often rated as suitable by human experts (x_10). Analysis results can be filtered by weaknesses or searched by terms, regardless of the user group (x_9). Quick-infos on selected terms in the analysis results are displayed by hovering over them (x_11), e.g. for the suggestion of *automated creating invoices* in Figure 8. Here, example-based explanations are included (x_7).

Actions, such as changing node labels or calculation formulas, as well as verifying and confirming identified outcome nodes of improvement measures or rejecting them if they are inappropriate, are performed in the *Action* area by system or domain experts (x_12). The edit icon opens an administration window for managing, adapting and extending the knowledge graph. During the training phase, domain experts can evaluate improvement measures according to the identified weaknesses of their result paths by approving them with *thumbs up* (+1), rejecting them with *thumbs down* (-1) or taking no action (evaluated with 0), in order to calculate weightings w as success rates (see Section 4.4). In live mode, the rating functions of the thumbs can be concealed from end users of the client side.

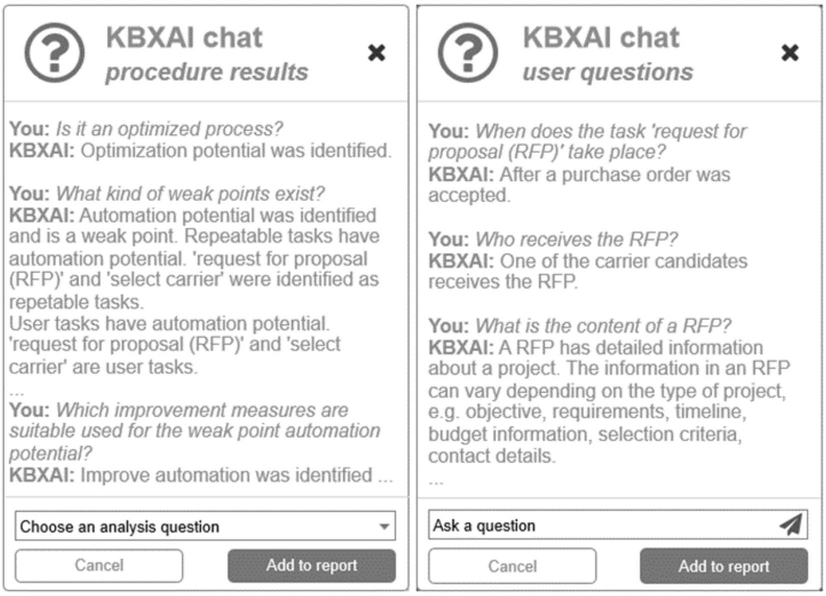

Fig. 10. Process analysis view: Chat-based explanations and user questions in the form of a QAD

The question mark icon is used to present analysis results in a Question-Answering-Dialog (QAD) (see Figure 10, left) and to ask individual user questions (see Figure 10, right). The chat-based explanation (x_4) of results is based on the entry nodes and deduction algorithms defined in the analysis procedure. Accordingly, the question *What kind of weak points exist?* is followed by the entry node *weak point* and the deduction algorithm *kind-of*, which represents the second analysis step (Q2) in the analysis procedure. Individual user questions are constructed using defined question modules. For instance, the question module *when* is used for chronological sequences of an activity. The question word *who* can be used to determine the process participants. The answers to descriptive questions, such as *What is the content of an RFP?*, are based on the domain knowledge of the analysis graph. The results are generated by calling the deduction algorithm *Characterize*, which analyzes all associated *has*- and *can*-relations. Furthermore, integration with language models is a potential avenue for addressing knowledge questions posed by users in a manner that is not process-instance specific.

The results of a process analysis can be summarized in the form of an analysis report, which is primarily aimed at end users (process owners, consultants or consulting clients) (see Figures 11). The report contains a comprehensive account of the analysis results with all path runs. As illustrated in the excerpt of Figure 11, the report begins with an overview of the analytical procedure employed (*What has been investigated?*). This is followed by a description of the individual analysis steps, accompanied by the respective path runs depicted in the graph. These are illustrated in the form of decision trees.

Figure 11 presents the decision tree of the analysis inference Q1 of the *Is-it* algorithm, with all path traversals in the graph displayed in numbered form according to the order in which the analyzed nodes were traversed. Upon determining three end events in the process model during path run number 13, the node *multiple endevents* was consequently activated (see path number 14 *true*). This resulted in the subsequent elements *complex process* and *weak point* also receiving the value *true*. Due to the modeled *Constraint* between weak point and optimized process ($r(v_{optimizedProcess}) = true \leftarrow r(v_{weakPoint}) == false$, see Section 4.1), which indicates that the node *optimized process* assumes the value false as soon as a weak point is detected, it is not an optimized process (see analysis result 2.1 in Figure 11).

The demonstrated process analysis was subjected to an evaluation in the context of a case study involving different experts. The findings are presented in the following chapter.

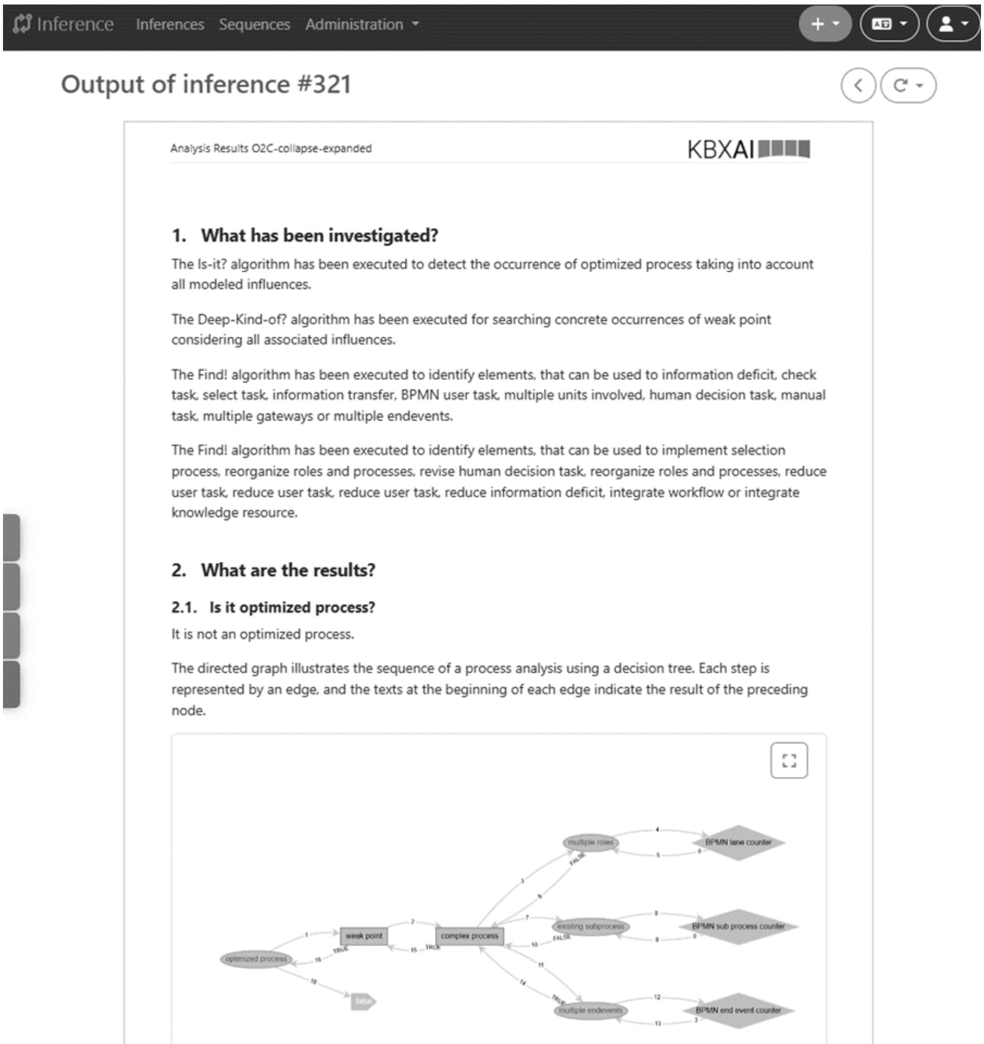

Fig. 11. Process analysis view: excerpt of analysis report

6. Evaluation of the XUI of KBXAI-PA

The evaluation of the designed and prototypically implemented XUI is carried out through several expert interviews with regard to selected design components. The demonstrated case study presented in the previous section is used for the interview subjects. As the evaluation of the XUI and its components is still at an early stage of research, the expert interviews are exploratory in nature. Compared to the first presentation of evaluation results [14], additional design components and evaluation criteria are now included, and the group of interviewees was extended. A broader evaluation with various case studies and a larger survey participation is forthcoming.

For the study, three consultants from a German SAP consulting company and six other participants from the fields of IT consulting, project management, logistics, and human resources were recruited. Four of the respondents had more than ten years of consulting experience, while three had less than five years in the consulting field. All participants carry out process analyses several times a month, or several times a week, for documentation purposes, as-is analyses, or to identify potential for improvement. Only four of the candidates have basic ML skills, while the other five have no ML skills.

All interviews consist of a brief introduction, a demonstration of the XUI using the case study above, testing of the XUI by the interviewee, and an evaluation of the XUI with a final discussion. In order to ensure a standardized evaluation of the interview results, the interviewees' opinions were rated using a five-point Likert scale as follows: 1: strongly agree, 2: rather agree, 3: neither, 4: rather disagree, or 5: strongly disagree. The results of the evaluation are presented below and the suggestions for improvement mentioned by the interviewees are summarized.

6.1. Relevance of design components

The visual explanation form is rated relevant by 100% of respondents, see Figure 12. More visual explanations, e.g. the illustration of process model extracts, should be taken into account in the further development of the XUI. Textual explanations, as largely implemented in the XUI prototype, also have a high approval rate of 67%. In contrast, the chat-based form of explanation was neither clearly approved nor completely rejected. The respondents were not sure whether this form of explanation could be relevant for the process analysis or not, which seems surprising in the age of generative chatbots. The reason for this could be attributed to the rather prototypical implementation of the KBXAI-PA chats. The presentation of multiple explanation views has been rated positively by all respondents, with 17% fully agreeing.

Local explanations that refer to a specific process analysis seem to be more relevant with 83% strongly agreeing than global explanations with general descriptions, which only receive 50% full agreement. Explanations that are supported by examples are considered relevant by the majority of respondents and should also be considered for the future.

Two-thirds of respondents believe that the relevance of features that lead to the identification of results, such as process deficiencies, should be displayed in the results overview, see *x*_9 in Figure 12. In the prototype, analysis results (weaknesses and improvement measures) can be listed according to the time of their identification or also according to the frequency of activated features (e.g. the weak point *complex process* can be activated by several indicators, such as several roles, several gateways, several end events or sub-processes). Search, filter, sort functions and control elements are considered very relevant by 83% of respondents. This emphasizes the need for a user-oriented presentation of analysis results. The accuracy indicator of an analysis result is classified as relevant in the same way (see *x*_10). In the prototype, the accuracy indicator for each improvement measure is indicated by the success rate, which shows the weighting according to user feedback in relation to the verification of the identified improvement measures. Quick-info via hover effect has also received a majority approval, whereby 33% of respondents categorized a quick info as neither relevant nor irrelevant. The correction of features or certain process

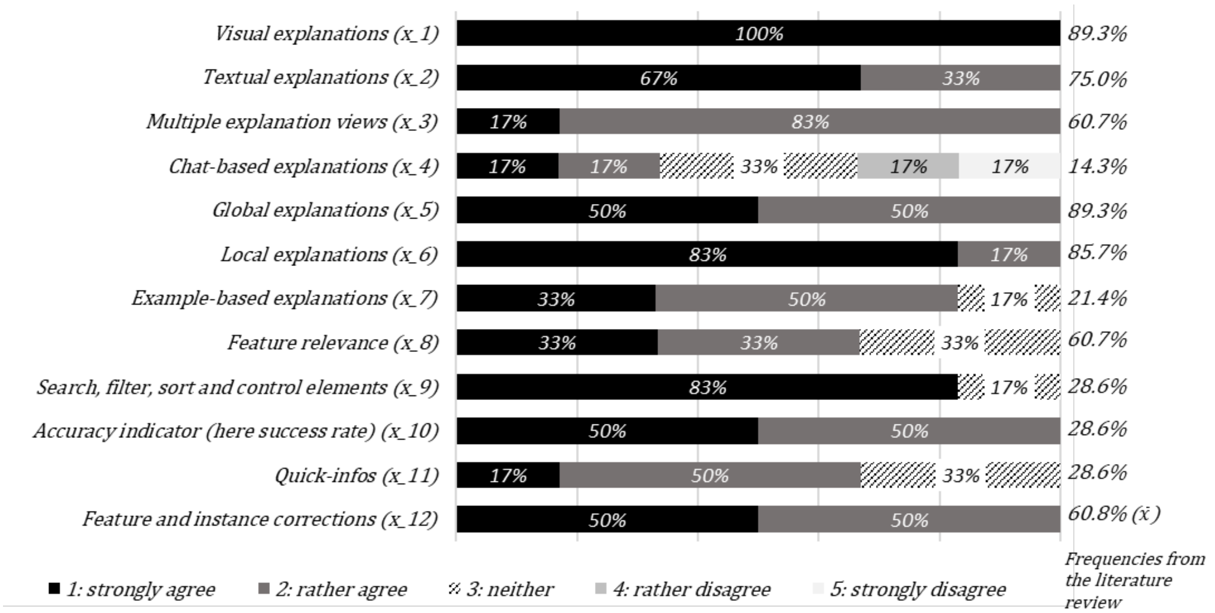

Fig. 12. Relevance of design components, extended in accordance with [14]

instances is sometimes considered very relevant and sometimes rather relevant (see *x*_12). This reflects the high interest of users in being involved in the explanation process of an XAI system in order to understand the results and be able to interact with the system. With regard to this matter, the results demonstrate the significance of incorporating human interaction when employing AI systems, with a view to overcoming algorithm aversion [6], and thus increasing the human acceptance of AI based processes [30].

Figure 12 shows the frequencies identified in the literature for the design components in addition to the exploratively determined ratings. When comparing the rating levels *strongly agree* and *rather agree* with the frequencies of the applied design components from the literature review, most of them show similar values. However, the low presence of example-based explanations in the literature (21.4%) in relation to their high relevance (83% agreement with levels 1 and 2) is noticeable. The paucity of example-based explanations in the analyzed XUIs can be explained by the necessity of adequate examples, ideally in ontological representation form, to establish how the results relate to such examples. However, none of the analyzed XUIs employed knowledge-based technologies, such as ontologies, to generate explanations. Likewise, basic interaction functionalities such as searching, filtering, sorting or control elements are rated with a relevance agreement of 83%, although they are only represented at 28.6% in the XUIs identified through the literature review. This tendency may be attributed to the fact that research-oriented evaluations of XUIs predominantly emphasize functionality and the investigation of hypotheses. Consequently, such common control elements tend to be ignored because the result itself, or an adequate explanation for a result, is often evaluated rather than the ability of the XUI to interact. User-oriented explanations, which become more comprehensible through examples and can be individualized through user interactions, are highly relevant and are still underrepresented in XUIs implemented to date. Accuracy indicators are also rarely represented in XUIs to date, but receive high to very high relevance ratings.

Nevertheless, discrepancies between the frequency of the components identified in the extant literature and the components deemed relevant by the respondents may also be due to the number of XUIs considered and the number and choice of experts interviewed.

6.2. Readiness for usage

The readiness to use the XUI of our KBXAI-PA approach is summarized in six statements, see Figure 13. A third of the respondents disagreed with the statement that they had doubts about the results. A minority of the respondents (less than half) stated that they were ambivalent regarding the statement. Nevertheless, 22% of the respondents expressed skepticism concerning the results of the XUI. The majority of respondents rated the forms of explanation implemented in the XUI and tested by the respondents (visual, textual, multiple views) as sufficient. Correction functions increase the willingness to use them, according to a majority of 83% of respondents, which is also reflected in the evaluation of the design components (see *x*_12 in Figure 12).

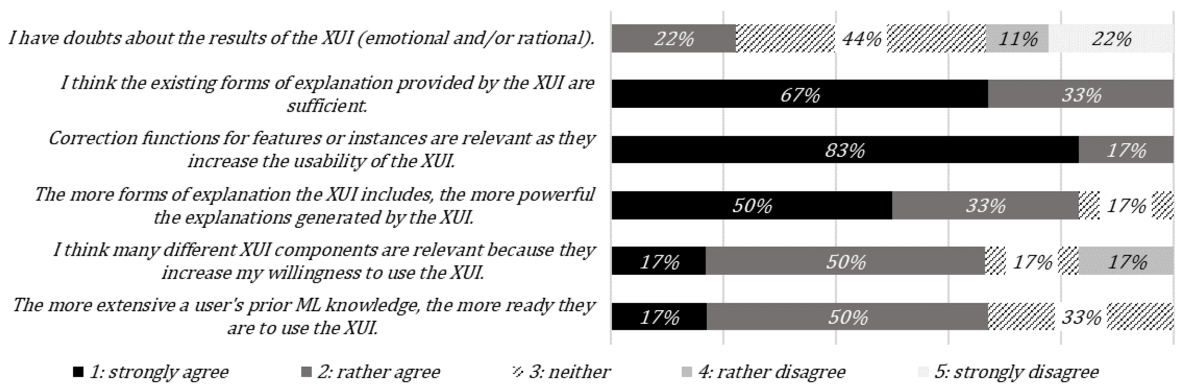

Fig. 13. Readiness for usage, extended in accordance with [14]

The number of explanation forms contained in an XUI seems to have a positive effect on the explanatory power. A total of 50% of experts provided a strong affirmative response to this statement, while 33% indicated a tendency to agree. However, a large number of different design components does not have the same effect on the willingness to use the XUI as different forms of explanation have on the explanatory power. 67% of respondents agree or strongly agree with the statement that many different XUI components increase the willingness to use, 17% neither agree nor disagree and a further 17% tend to disagree with this statement.

While a total of 17% of respondents expressed full agreement with the statement that ML knowledge increases the willingness to use it (see last statement in Figure 13), approximately one-third of respondents indicated ambivalence on this matter. Furthermore, respondents noted that familiarity with the fundamental analysis procedures of KBXAI-PA fosters confidence in the outcomes and facilitates targeted interaction and evaluation of the analysis results for the interactive learning mechanisms. From the perspective of end users and domain experts, it can be posited that general ML knowledge facilitates the intended and effective use of the XUI for process analysis.

In this context, it can be posited that the provision of correction options constitutes a fundamental functionality that not only serves the continuous enhancement of the XUI of the KBXAI-PA model by a human user, but also amplifies the user's readiness to utilize the system.The implementation of diverse forms of explanation has been shown to enhance explanatory power.The design components implemented in the presented XUI appear to be adequate for the participants of the tool demonstration. The results of the expert interviews suggest that the number of implemented design components is not a determining factor in the users' willingness to use the tool. Instead, a targeted selection of suitable components, tailored to the respective application domain, emerges as a decisive aspect.

6.3. General properties and discussion results

All participants consider the XUI to be intuitive and user-friendly, see Figure 14. Over half of the respondents rated the XUI as rather explainable, of which 22% even fully agreed. In contrast, the majority of respondents (67%) agreed with the interpretability, 11% of the respondents expressed their full conviction. This opinion is also reflected in the responses to the following statements: compared to the assessment of explainability, two-thirds of the respondents agree with a transparent (comprehensible) XUI, while no respondents expressed disagreement. Analogous to interpretability, 22% of respondents consider the XUI to be trustworthy and 45% tend to agree with this statement. The interactivity of the XUI is confirmed by 77% of respondents, reflecting the interactions implemented.

According to the expert interviews, the developed XUI provides explainable and technically comprehensible results in a user-friendly manner, while interpretability and trustworthiness are considered to be less fulfilled. This result is positive for the developed XUI insofar as the user interface reveals the logic of the resulting analysis results. The expressiveness of the generated results is mainly influenced by the scope of the knowledge graph, which can be extended or optimized by domain-specific knowledge.

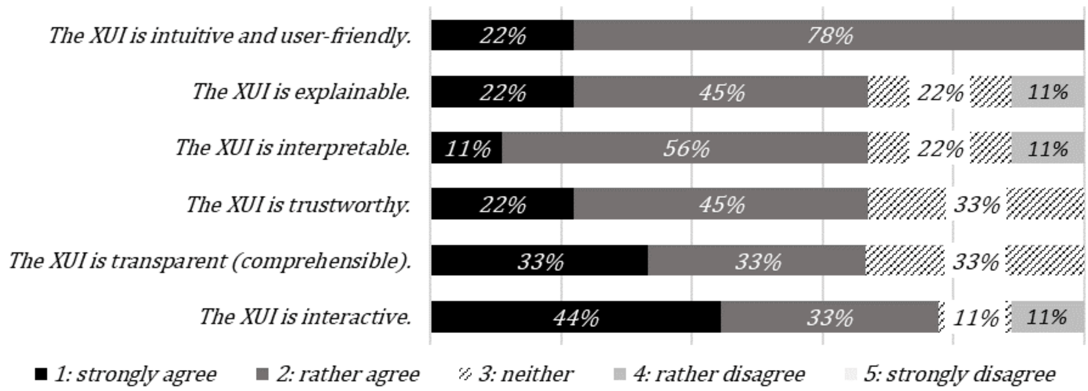

Fig. 14. General properties, extended in accordance with [14]

Additionally, the experts suggested that the analyzed BPMN models should be presented in excerpts in the results report. Another wish that was expressed was the connection of the analysis tool to existing source systems of process data as well as integration options with other analysis tools, such as process mining applications, e.g. the EMS from Celonis. At the end of the evaluation, the participants were asked whether they could imagine using the XUI in their daily work. Five of the interviewees answered in the affirmative and stated that the XUI would be interesting for testing processes in the future, that the XUI could make daily work easier and that the XUI would be helpful for process analysis and recording in order to identify potential for improvement. The other four interviewees were still undecided, partly due to the fact that process models in the specification language BPMN are required for the process analysis procedure. In order to analyze processes based on event log data, BPMN models can also be generated and used automatically by using an API to a process mining tool.

7. Conclusion and research perspectives

The knowledge graph-based approach to explainable process analysis (KBXAI-PA) represents a hybrid AI approach that combines symbolic methods of knowledge representation with machine deduction algorithms and generates user-driven and understandable results while taking human interactions into account. Interpretable models make it possible to reconstruct the origin of analysis results in the form of result paths. In order to make the system behavior transparent and comprehensible for users, AI results are presented in a readable, understandable and plausible way in an explanation user interface (XUI). The interactive learning method enriches the knowledge graph with user feedback, which on the one hand leads to the adaptation of the analysis procedure and on the other hand continuously refines process analysis results. Verified results are taken into account and prioritized by the deduction algorithms. Only if the system behavior and the results of AI systems are understandable for human users can interactive learning models be trained in the best possible way through user interactions such as confirmations, rejections and corrections. The acceptance of analysis results and their use in decision-making processes increases if the plausibility can be verified by external feedback [25].

For the demonstration and evaluation of human interactions for the knowledge graph-based XAI approach, an XUI was developed using a design catalog [14]. For this purpose, various XUI components were selected according to their relevance and developed for an XUI in the form of a prototype. In summary, it can be stated that different forms of explanation can support the explanatory power in the sense of comprehensible results. However, the highest possible number of implemented design components does not necessarily increase the willingness to use AI systems with an XUI. Users can be overwhelmed by the large number of explanation components on offer. Interaction functions that contribute to both the explanation and adaptation of the learning model are evaluated positively and can increase the readiness to use the system. Interaction and correction elements give the user the feeling of participating in the behavior of the system and of being able to continuously improve the analysis basis, the knowledge graph, through their own domain knowledge, which can increase user acceptance [8] and prevent algorithm aversion [6].

The evaluation of explanation components of an XUI should always be considered with regard to the respective implementation and case studies carried out, as the design of an XUI can vary depending on the type of AI system and design requirements. Usability aspects can influence the evaluation of XUIs. The design of XUIs has a considerable influence on how analysis results and their interpretations are communicated to and understood by users and how users can react to system results through review and interaction. Follow-up evaluations in larger survey rounds with a quantitative study design and consideration of different user perspectives are currently in progress.

In addition, the suitability of XUIs for improving interactive learning methods needs to be investigated. In addition to its explanatory power, an XUI should also contribute to improving interactive learning models. In this way, learning models can be aligned with human thinking and analysis results can be trained both transparently and comprehensibly depending on the application domain.

The evaluation of the interpretability of XAI results can be advanced through the analysis of syntactic, pragmatic, and epistemic properties [3] of explanations in subsequent evaluations. In light of a philosophy of science background, it is worthwhile to discuss in this context how XAI results can be combined and enriched by metaphorical or unificatory explanations to increase their ontological explanatory power [18].

With regard to the use of BPMN models from process mining analyses, a further evaluation can be carried out by investigating the extent to which process mining analyses can be expanded and improved through integration with KBXAI-PA and how useful the generated process analysis results are in practice.

Self-service applications for AI-based process analysis with user-centric XUIs should be offered on digital platforms not only for consultants, but also for consulting clients to self-analyze their own business processes. A reduction in the required number of consultants for process analysis through automation leads to cost advantages as well as increased productivity and thus to improved quality assurance towards clients. Moreover, automated and, thus, more cost-effective analysis services, can open up new client segments, who are not able or willing to pay high rates for individualized human consulting services today.

References

[1] S. Amershi, M. Cakmak, W.B. Knox and T. Kulesza, Power to the People: The Role of Humans in Interactive Machine Learning, *AI Magazine* **35**(4) (2015), 105–120.

[2] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila and F. Herrera, Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Information Fusion* **58** (2020), 82–115.

[3] C. Bauckhage, J. Fürnkranz and G. Paaß, Vertrauenswürdiges, transparentes und robustes Maschinelles Lernen, in: *Handbuch der Künstlichen Intelligenz*, G. Görz, U. Schmid and T. Braun, eds, Walter de Gruyter GmbH, Berlin, 2021, pp. 571–600. ISBN 9783110659948.

[4] A. Bibal and B. Frénay, Interpretability of Machine Learning Models and Representations: an Introduction, in: *Proceedings of the 24th European Symposium on Artificial Neural Networks (ESANN)*, 2016, pp. 77–82. https://www.researchgate.net/profile/adrien-bibal-2/publication/326839249_interpretability_of_machine_learning_models_and_representations_an_introduction.

[5] M. Chromik and A. Butz, Human-XAI Interaction: A Review and Design Principles for Explanation User Interfaces, in: *INTERACT 2021. LNCS*, Vol. 12933, C. Ardito and et al., eds, Springer, Cham, 2021, pp. 619–640. ISBN 978-3-030-85615-1.

[6] B.J. Dietvorst, J.P. Simmons and C. Massey, Overcoming Algorithm Aversion: People Will Use Imperfect Algorithms If They Can (Even Slightly) Modify Them, *Management Science* **64**(3) (2018), 1155–1170. doi:10.1287/mnsc.2016.2643.

[7] J. Dodge, S. Penney, C. Hilderbrand, A. Anderson and M. Burnett, How the Experts Do It, in: *Engage with CHI*, R. Mandryk, M. Hancock, M. Perry and A. Cox, eds, The Association for Computing Machinery, New York, New York, 2018, pp. 1–12. ISBN 9781450356206. doi:10.1145/3173574.3174136.

[8] L. Drewes and V. Nissen, Akzeptierte Geschäftsprozesse gestalten und implementieren, *HMD* **59**(2) (2022), 572–587. doi:10.1365/s40702-022-00856-x.

[9] J.J. Dudley and P.O. Kristensson, A Review of User Interface Design for Interactive Machine Learning, *ACM TiiS* **8**(2) (2018), 1–37.

[10] M. Eiband, H. Schneider, M. Bilandzic, J. Fazekas-Con, M. Haug and H. Hussmann, Bringing Transparency Design into Practice, in: *23rd Int. Conference on Intelligent User Interfaces*, S. Berkovsky, ed., ACM Conferences, 2018, pp. 211–223. ISBN 9781450349451.

[11] J.A. Fails and D.R. Olsen, Interactive machine learning, in: *Proc. of the 8th Int. Conference on Intelligent User Interfaces*, ACM Press, New York, USA, 2003.

[12] A. Füßl and V. Nissen, Interpretability of Knowledge Graph-based Explainable Process Analysis, in: *IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Laguna Hills, CA, USA*, 2022, pp. 9–17. ISBN 2831-7203. doi:10.1109/AIKE55402.2022.00008..

[13] A. Füßl, V. Nissen and S.H. Heringklee, Interactive Machine Learning of Knowledge Graph-Based Explainable Process Analysis, in: *Advanced Information Systems Engineering Workshops*, M. Ruiz and P. Soffer, eds, Lecture Notes in Business Information Processing, Springer International Publishing AG, Cham, 2023, pp. 112–124. ISBN 978-3-031-34984-3. doi:10.1007/978-3-031-34985-0_12.

[14] A. Füßl, V. Nissen and S.H. Heringklee, An Explanation User Interface for a Knowledge Graph-Based XAI Approach to Process Analysis, in: *Advanced Information Systems Engineering Workshops*, J.P.A. Almeida, C. Di Ciccio and C. Kalloniatis, eds, Lecture Notes in Business Information Processing, Springer Nature Switzerland and Imprint Springer, Cham, 2024, pp. 72–84. ISBN 978-3-031-61003-5. doi:10.1007/978-3-031-61003-5_7.

[15] A. Füßl, V. Nissen, S. Dopf and F.F. Füßl, An Inferential Knowledge Model for the Digitalisation and Automation of Business Process Analysis, in: *Proc. Wirtschaftsinformatik*, N. Gronau, M. Heine, K. Poustcchi and H. Krasnova, eds, GITO, 2020, pp. 185–200. ISBN 9783955453367.

[16] F.F. Füßl, *Entwicklung eines Modells zur Anwendung inferenzfähiger Ontologien im Software Engineering*, Doctoral dissertation, TU Ilmenau, 2016.

[17] B. Ghai, Q.V. Liao, Y. Zhang, R. Bellamy and K. Mueller, Explainable Active Learning (XAL), *Proc. of ACM on Human-Computer Interaction* **4**(CSCW) (2021), 1–28.

[18] G. Guizzardi and N. Guarino, Explanation, semantics, and ontology, *Data & Knowledge Engineering* **153** (2024), 102325:1–18. doi:10.1016/j.datak.2024.102325.

[19] D. Gunning and D. Aha, DARPA's Explainable Artificial Intelligence (XAI) Program, *AI Magazine* **40**(2) (2019), 44–58.

[20] J.L. Herlocker, J.A. Konstan and J. Riedl, Explaining collaborative filtering recommendations, in: *Proc. of ACM conference on Computer Supported Cooperative Work*, W. Kellogg, ed., ACM Conferences, 2000, pp. 241–250. ISBN 1581132220.

[21] F. Hohman, A. Srinivasan and S.M. Drucker, TeleGam: Combining Visualization and Verbalization for Interpretable Machine Learning, in: *2019 IEEE Visualization Conference (VIS)*, IEEE, Piscataway, NJ, 2019, pp. 151–155. ISBN 978-1-7281-4941-7. doi:10.1109/VISUAL.2019.8933695.

[22] A. Holzinger, Interactive Machine Learning, *Informatik-Spektrum* **39**(1) (2016), 64–68. doi:10.1007/s00287-015-0941-6.

[23] T. Kulesza, M. Burnett, W.-K. Wong and S. Stumpf, Principles of Explanatory Debugging to Personalize Interactive Machine Learning, in: *Proc. of 20th Int. Conf. on Intelligent User Interfaces*, O. Brdiczka, ed., ACM, 2015, pp. 126–137. ISBN 9781450333061.

[24] B.Y. Lim and A.K. Dey, Assessing demand for intelligibility in context-aware applications, in: *Proceedings of the 11th international conference on Ubiquitous computing*, S. Helal, ed., ACM Conferences, ACM, New York, NY, 2009, pp. 195–204. ISBN 9781605584317. doi:10.1145/1620545.1620576.

[25] C. Meske, E. Bunde, J. Schneider and M. Gersch, Explainable Artificial Intelligence: Objectives, Stakeholders, and Future Research Opportunities, *ISM* (2020), 1–11.

[26] M. Millecamp, N.N. Htun, C. Conati and K. Verbert, To explain or not to explain, in: *Proce. of the 24th Int. Conference on Intelligent User Interfaces*, W.-T. Fu, ed., ACM Conferences, ACM, New York, NY, 2019, pp. 397–407. ISBN 9781450362726.

[27] J.D. Moore and C.L. Paris, Requirements for an expert system explanation facility, *Computational Intelligence* **7**(4) (1991), 367–370.

[28] F. Morelli, S. Geschwill, K. Zerr and C. Lossos, Rationalität maschineller Entscheidungen im Unternehmen durch die Einbindung von Explainable Artificial Intelligence (XAI), *Tagungsband 33th AKWI* (2020), 8–17.

[29] N. F. Noy and D. L. Mcguinness, Ontology Development 101: A Guide to Creating Your First Ontology (2001). https://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

[30] V. Nissen and T. Müllerleile, How to Design, Implement, and Manage Accepted Business Processes, *Advances in Consulting Research* (2019), 147–170. doi:10.1007/978-3-319-95999-3_7. https://link.springer.com/chapter/10.1007/978-3-319-95999-3_7.

[31] V. Nissen, N.s. Joksch and A. Füßl, Wie die Corona-Pandemie die Unternehmensberatung verändert hat, in: *Gestaltung des Wandels im Dienstleistungsmanagement*, M. Bruhn and K. Hadwich, eds, Springer, Wiesbaden, 2023, pp. 633–670. ISBN 978-3-658-41813-7.

[32] A. Páez, The Pragmatic Turn in Explainable Artificial Intelligence (XAI), *Minds and Machines* **29**(3) (2019), 441–459. doi:10.1007/s11023-019-09502-w.

[33] K. Peffers, T. Tuunanen, M.A. Rothenberger and S. Chatterjee, A Design Science Research Methodology for Information Systems Research, *JMIS* **24**(3) (2007), 45–77.

[34] B. Shneiderman, Bridging the Gap Between Ethics and Practice, *ACM Transactions on Interactive Intelligent Systems* **10**(4) (2020), 1–31.

[35] W.M.P. van der Aalst, *Process Mining: Data science in action*, 2nd edn, Springer, Berlin, Heidelberg, New York, Dordrecht, London, 2016. ISBN 9783662498514.

[36] J. Webster and R.T. Watson, Analyzing the Past to Prepare for the Future: Writing a Literature Review, *MIS Quarterly* **26** (2002).

[37] W. Xu, Toward human-centered AI, *Interactions* **26**(4) (2019), 42–46.