Design Decision Framework for AI Explanations

Oghenemaro Anuyah
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
oanuyah@nd.edu

William Fine
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
wfine@nd.edu

Ronald Metoyer
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
rmetoyer@nd.edu

ABSTRACT

Explanations can help users of Artificial Intelligent (AI) systems gain a better understanding of the reasoning behind the model's decision, facilitate their trust in AI, and assist them in making informed decisions. Due to its numerous benefits in improving how users interact and collaborate with AI, this has stirred the AI/ML community towards developing understandable or interpretable models to a larger degree, while design researchers continue to study and research ways to present explanations of these models' decisions in a coherent form. However, there is still the lack of intentional design effort from the HCI community around these explanation system designs. In this paper, we contribute a framework to support the design and validation of explainable AI systems; one that requires carefully thinking through design decisions at several important decision points. This framework captures key aspects of explanations ranging from target users, to the data, to the AI models in use. We also discuss how we applied our framework to design an explanation interface for trace link prediction of software artifacts.

CCS CONCEPTS

• **Human-centered computing → HCI design and evaluation methods**.

KEYWORDS

Explainable AI, interaction design, design activity, artificial intelligence, machine learning, user research, design decision

1 INTRODUCTION

Artificial Intelligence (AI) is increasingly becoming a standard technology in computational systems. Today, AI-based solutions are prevalent in industries such as healthcare [25], education [31], food technology [14], and financial services [20], among others. As the use of AI continues to grow, it is critical that intelligent computational systems interact with human partners to address complex problems. An important part of this interaction is the ability of an AI system to be able to "explain" it's decision to human collaborators and users. In human-AI interaction, an effective explanation can help users gain new insights about the problem an AI model intends to solve, facilitate trust and understanding in AI systems, and assist users in making informed decisions [10, 22]. Hence, proper investigation on how to adequately design explanations for AI systems is a must.

In the educational literature, an explanation is a tool used by a speaker to foster understanding or to "give sense" to the object of communication or a discussion [24]. The *object* of communication is known as *what* the speaker (or *author*) aims to explain, which may be a concept about a topic of interest, a mistake, or an algorithm for example. The goal of an explanation is to make clearer the meaning of the target object to a target audience. To achieve this goal, three important parameters to consider during an explanation dialogue are: *the explanation object*, *the mode of communicating the explanation* and *the target audience of the explanation.* In the context of AI, we refer to explanations as "*a tool/an interface used by a designer/researcher to foster understanding or to "give sense" to an AI model's decision.*" Here, the explanation *object* is the AI algorithms' decision, the *author* is the designer or researcher that designed the explanation, and the *target audience* are the users that the explanations are designed for or who will be interacting with the explanation. Similarly, the three parameters that researchers should consider for AI explanation design are: *what to explain* (explanation object), *who to design the explanation for* (target user), and *how to present the explanation* (mode of explaining).

Research on the design of explainable AI (XAI) has received significant attention over the years [10, 11, 21]. A number of researchers in the XAI area, however, do not explicitly consider all three explanation design parameters in tandem [15, 19]. Several explanation designs make assumptions about who the target users are and what they need, as opposed to designing for the actual needs of the intended explanation audience [19]. Further, a number of existing studies assume how to represent the explanations (e.g., through the use of interactive visualizations [15] or free text/natural language [8]), ignoring the fact that users have different needs and capabilities. Unfortunately, failure to tailor the explanation design to target users may render the explanations ineffective, potentially risky, and under-used in real world scenarios [10].

Motivated by these concerns and with the aim to promote more human-centered explanation interface designs, in this paper, we propose a design decision framework to guide designers of explanation interfaces throughout the design process. This framework was inspired by the nested model for visualization design introduced by Munzner et al. [23]. The framework is intended to help designers to consider the most relevant aspects in their design decisions and to validate those design decisions as appropriate.

Mensch und Computer 2021, Workshopband, Workshop on User-Centered Artificial Intelligence (UCAI '21)

https://doi.org/10.18420/muc2021-mci-ws02-237

Our framework consists of three phases, carefully chosen to ensure that designers understand how an AI problem intersects with the user needs, consider the explanation goals and users' tasks, identify the relevant data, and select appropriate representations for the explanations. We also discuss threats at each phase that can invalidate the design, as well as suggestions on how to address these threats.

2 EXISTING EXPLANATION SYSTEMS DESIGN

We analyzed a number of existing explanation system designs published in the research literature between 2016 and 2021. We began our search on publication databases such as Google Scholar, ACL Anthology, and ACM Digital library by searching for the keywords "explanations", "explanations design" and "AI interpretation design". Based on this selection criteria, we identified 20 research articles. We have reviewed the selected papers with the aim to identify the space of design decisions for explanations, and the extent to which target users play a role in the design and evaluation process. A summary of our analysis can be accessed here: https://design-decision-xai-framework.github.io/pages.

We have observed that more than 90% of the research studies that we have reviewed (including those that we did not discuss directly in this paper) do not explicitly consider target users throughout the explanation design process. As discussed by [22], a number of explanation designs are inspired from the research literature, as opposed to being inspired by *who* would be interacting with the design. Common explanation designs are characterized either based on the model task (prediction or classification) or the input type (text or image), largely ignoring the user side of the problem. Moreover, very few of these examined studies evaluated the explanation designs with the target users to gather insights on whether the explanations addressed their needs [2, 26]. When users needs are not properly considered, it is quite possible that the resulting explanation may not support them appropriately. With the aim to address this issue, we propose a design decision framework that researchers can utilize throughout their explanation design process to make and validate their design decisions.

3 DESIGN DECISION FRAMEWORK

From our review of the literature on explanation system designs, we identify potential weaknesses in prior designs–mainly due to the fact that users are not at the core front of explanation designs. We use these existing systems to develop and illustrate our framework which is inspired by Munzner et al. [23]; who proposed a nested model for designing visualizations. Our design decision framework consists of three phases which we discuss in the following sections.

3.1 Domain characterization

Typically, intelligent systems are developed to solve a specific problem (e.g., classifying a data point or recommending an item to a user) for a specific user. An effective explanation should take into consideration both the problem that the intelligent system aims to solve and the user [30]. Throughout this paper, we will interchangeably use explanation audience or target user to refer to users for which the explanation is designed.

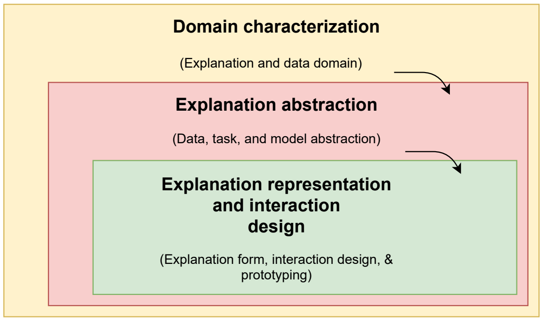

Domain characterization
(Explanation and data domain)
Explanation abstraction
(Data, task, and model abstraction)
Explanation representation and interaction design
(Explanation form, interaction design, & prototyping)

Figure 1: Overview of the explanation design decision framework. Decisions in upstream of the framework affects decisions made by designers later. Details in "()" indicate important concepts to consider at each phase.

The first component of the XAI design decision framework entails understanding the intelligent system's domain and the needs of the target users. Two questions that explanation designers should aim to answer here are: (i) who are the target users and (ii) what questions do these users need answered based on not just the intelligent system, but also the general problem space the system is developed to address. In Munzner's nested model, this is referred to as [23] – domain problem and data characterization, and entails gaining and understanding of the domain and data that the visualization will be designed for.

We observe in the literature that users of intelligent systems ask questions about both the **data domain** as well as questions about the **explanation domain** itself. The data domain refers to data that is relevant to the problem space itself. For illustration purposes, we will use the outfit recommendation explanation system designed by Lin et. al [18] as an example for our discussion. In this paper, an intelligent system was designed to recommend outfits to customers and the author's aim was to explain the system's output. Here, the problem space is "clothes outfits" in general, while the intelligent system's task is to "recommend outfits", with the user's end goal being to make a purchase. The data domain then, is "clothing outfit" and the explanation domain is "recommendation."

Hence, questions pertaining to the data domain should capture user's needs and expectations around the problem space more broadly. For instance, some questions that users may generally ask about outfits are *what color combinations usually work well for clothing?*, *what is this clothing style?*, and *what clothing is appropriate for the summer?*. On the other hand, the explanation domain refers to data pertaining to the intelligent system's decision or output. For example, users might ask *who are other users that have rated or bought this outfit?*, or *what are the specific features of this outfit that makes this system recommend it to me?* The designer's goal, therefore, is to answer questions not only about the explanation, but also questions about the data domain that may be relevant in understanding the explanation.

We argue that separating the data domain from the explanation domain is imperative. Understanding an explanation may require an in-depth understanding of the data domain. In the clothing recommender, for example, the authors primarily provide explanations of the recommendation model's decision (i.e., explanation domain). Users, however, may need further information about the clothing domain to understand those explanations. An individual that is more experienced in fashion may very well understand why a set of recommendations is reasonable with regards to the style and color combination. An individual without that experience, however, could benefit from knowing what are good "color combinations" for clothing.

In summary, we observed in the literature that explanations sometimes do not adequately address potential user needs and that users may be left with more questions in mind as they interact with an explanation. This stems from the fact that the explanation may not capture information that they initially expected to see or may not align with their knowledge level (e.g. with respect to either the data or problem domain) or their capabilities [21, 22, 29].

To avoid this problem, the desired outcome of this component of the framework is a set of questions obtained directly from users that should capture information that they expect to see in explanations. Designers can obtain this information by leveraging user research methods that involve speaking directly with representative users such as through interview sessions, focus groups, or contextual inquiries, among others. In addition, designers should identify the data necessary to answer those questions.

3.1.1 Threat to validity. As discussed earlier, considering the threat to validity at each phase of the design helps to identify ways to validate the design decisions made by the researcher throughout the design process. Given that this phase of the framework entails understanding target users and identifying questions that capture their needs, one threat to validity here would be *addressing the wrong question(s)*. Consequentially, addressing the wrong question(s) would translate to designing explanations that may end up being irrelevant, non-useful, or not needed at all. Therefore, it is imperative that designers do not dwell on assumptions (e.g., personal interests or explanation strategies based on prior designs in the research literature [22]), as this may introduce bias to the design. Designers should validate design decisions in this phase by engaging with target users through interviews or observational studies in order to verify that the right problem or questions are being addressed. Another threat to validity at this phase is not distinguishing the data from the explanation domain, when gathering user questions. A detriment of not doing so may lead to designers focusing on one of these domains, most likely the explanation domain, whilst ignoring opportunities for data domain related questions. To mitigate this threat, researchers should make sure to consider these two domains separately and gather user questions that pertain to both.

3.2 Explanation abstractions

Formative user studies will result in an understanding of domain specific (data and explanation domains) questions and data. While useful, it is sometimes difficult to map these domain specific needs and questions to explanation representations from the literature that were designed for other specific domains. As in many computing problems, it is important to attempt to move from domain-specific problems to abstract problems for which we may already know useful solutions.

From the previous phase (section 3.1), it is expected that researchers would already have determined the target audience and domain-specific questions that they would like to see addressed in the explanation interface. Researchers should therefore endeavor to determine the intermediate generic/abstract tasks that these users should be able to perform with the explanation. Taking inspiration from Munzner [23], this can be accomplished by mapping the set of domain-specific questions obtained from users, to more abstract or generic tasks in the vocabulary of AI/ML. This is known as **task abstraction**.

Designers may observe that some of the user questions related to the explanation domain indicate that they may want to *identify important features* that influence the intelligent system's decisions or *compare outputs* generated by the system after tweaking one or more of the attributes in the system's input. A review of literature in the XAI area have revealed a number of common abstract tasks for explanations (See Table 1). It is important to note that multiple questions can map to the same abstract task. In terms of user questions related to the data domain, a useful explanation here would be one that helps the user understand the data domain better. Hence, abstracting these questions would involve helping users to *identify definitions or concepts* related to the data domain.

Designers should also transform raw data of the problem domain (as discussed in section 3.1) into abstract data types, as this would be useful for determining suitable representations for the explanations. This is known as **data abstraction** [23]. In general, common data types are qualitative/text or quantitative/numerical. For example, in the explanation system designed by Gehrmann et al. [11], the authors displayed attention words that the model focused on to generate its output. Here, these attention "words" are of the qualitative/text data type. Designers must also consider derived data for the explanation. One difference between visualization design, for example, and explanation design, is that the intelligent system itself may represent a form of derived abstract data type (e.g., a decision tree, a classification decision boundary, or parameters in a regression model). In this case, a derived abstract data type is not necessarily based on the result of the output of the model, or the raw data used for training the AI model. This data may be derived as a result of the model's training process and informs the decision made by the model. For instance, the weights of a regression model or the decision tree itself can be considered as a form of derived data. Presenting the derived data to users may help them gain insights on the inner-workings of the model. Hence, **model abstraction** is important to consider in explanation design.

In a nutshell, the outcomes of this phase of the design decision framework would be a collation of abstract tasks to be performed by users. Additionally, designers should transform raw data for answering user questions into data types that are useful for representing the explanations, and also generate derived data types for the model.

3.2.1 Threat to validity. There are two primary threats to validity at this phase of the framework; (i) choosing abstract tasks that do not align with user questions identified in section 3.1 and (ii) selecting / using the wrong data types for data instances that are relevant for designing the explanations, whether derived or not. To validate these threats, designers should ensure that target users are

Table 1: Examples of task abstractions. Tasks related to the explanation domain were identified from the literature. We added tasks related to the data domain.

Explanation domain task abstractions
Compare different features [3, 13, 30]
Compare input with output [11]
Compare different outputs / outcomes [18]. Users may do so to determine reasons why some outputs are of more value than others [30]
Identify attention features [11, 28, 30], i.e., indicating the input features of the model that is important or identifying if it has a positive or negative influence [30]
Determine relevance [13, 16] i.e., contribution of features in the model's decision or overall performance
Determine causation [4, 16, 30], i.e., understand why certain objects or inputs are considered similar or different
Discover insights about the model's internals [3, 32]
Examine alternative predictions [30]
Relate features [1]
Data domain task abstractions
Identify data domain concepts or definitions

involved in evaluative studies in the form of controlled user studies, anecdotal evidence of utility for real users, and/or longer term field studies.

3.3 Explanation representation and interaction design

This component of the design decision framework involves generating **explanation representations** and creating the **interaction design** of the explanation interface. In the research literature, explanations are commonly represented as interactive visualizations (e.g., partial dependence plots [15], saliency heat map or attention scores [1, 8, 28], decision rules [17], tuples or graphs [12], and feature attribution or influence scores [5, 32]), representative training examples [7], semantic concepts [34], and free text (i.e., natural language) [8] among others. There is, however, limited justification as to why and how these explanation types are applicable to the problem the intelligent system aims to solve, relevant to the target users' explanation goal, or even useful to them.

Given the user tasks and data types identified in section 3.2, designers should identify explanation representations that are specifically suited to these data and tasks. As discussed in [29], it is possible that while certain explanation representations are presumed to be the most applicable for solving certain problems (e.g., saliency map visualizations for neural image classification representations), these may actually decrease a user's ability to understand the explanation [6]. Therefore researchers should ensure that the selected explanation representations: (i) are comprehensible or understandable by the target users and (ii) answers their questions or supports their tasks. At this phase, design decisions should be based on prior literature and/or formative user research. Prior research has identified a set of representations that are particularly useful for specific tasks and data types. For new task and data type combinations, researchers are encouraged to utilize formative methods that involve co-designing with potential target users.

In addition to selecting appropriate representations in support of the user's abstract tasks, at this phase, researchers should also design the interaction between all explanation components in a way that is meaningful to the user and that provides a good experience for them. In this case, we consider each of these individual representations to be an *explanation component*, as it captures aspects for explaining the intelligent system's decisions to the target user. For instance, if one of the abstract tasks is *identify attention features* and researchers have determined that the most applicable representation is a feature attribution visualization coupled with free text/natural language, this on its own is an explanation component. To the best of our understanding, there are no existing studies on selecting the best interaction style (e.g., linking, animations, etc.) between explanation components. Researchers should, therefore, study ways on adequate interaction design of the explanations. For example, work on narrative visualization and methods for linking texts and visuals may be appropriate for explanation design [27, 33]. Researchers should also decide how to arrange the information and explanation components on the explanation interface screen.

The outcome of this phase of the design decision framework should be a functional prototype that target users can interact with and provide feedback on whether the interaction design of the explanations captures their desired expectations and capabilities.

3.3.1 *Threat to validity.*

At this phase, there are two primary threats to validity: (i) the explanation representation does not effectively support the abstract tasks given the user's needs and capabilities, and abstract data type, and (ii) designers do not justify their explanation interaction design choices. To mitigate the first threat, it is recommended that researchers: (i) consult and understand the literature on appropriate explanation representations for specific data and tasks, and (ii) conduct summative user studies with potential end users. For the latter, ideally, we recommend that designers should validate this threat by consulting the research literature on appropriate interactions for explanation interfaces. However, this is still an open problem in the XAI area as to the best of our understanding, there are no prior works that have explored this aspect. Hence, there opens up an opportunity for researchers in the XAI area to explore.

Table 2: Examples of task abstractions identified based on software traceability in the healthcare system. "Exp" refers to questions related to the explanation domain, while "Data" refers to questions pertaining to the data domain.

User questions	Domain	Abstract tasks
What are the types of the two linked artifacts?	Exp	**Identify** artifact types
What concepts do the two artifacts have in common?	Exp	**Compare** artifact content
What concepts does the system consider in determining the trace links?	Exp	**Identify** attention features
What does [concept X] mean?	Data	**Identify** data definitions
What is the semantic relationship between [concept X] and [concept Y]?	Exp	**Relate** features
How confident is the intelligent system in the link prediction? [Predicted score]	Exp	**Determine** relevance

Table 3: Examples of data relevant for answering user questions, along with their respective abstract data types.

Data	Abstract data type
Artifact content	Qualitative (source code, structured text, descriptive text)
Concept definitions	Qualitative (text)
Concept relationships (**derived**)	Qualitative (semantic relationships) **[determined from ontology]**
Similarity score between the source and target artifacts	Quantitative
Concept importance	Quantitative
Link confidence score	Quantitative

4 APPLICATION EXAMPLE

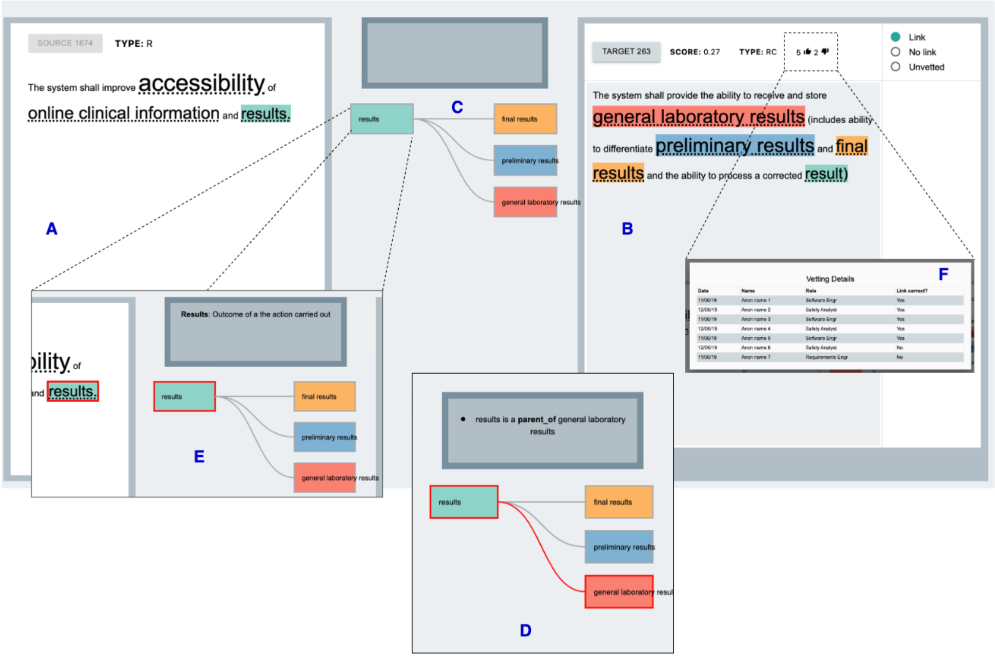
In a recent study that we conducted (haven't been published yet), we created an explanation interface to support the prediction of trace links between software artifacts. Examples of software engineering artifacts include code, design documents, requirements, and bug reports. A team of machine learning experts designed an algorithm that leverages information from an ontology, in order to predict the extent to which two software artifacts should be linked. In this case, a high prediction score indicates that two compared software artifacts are semantically related. The aim of the design research team then was to provide a rationale for why two artifacts have been predicted to be linked by the AI model.

We carried out a co-design activity with a team of experts (n=3) working in the traceability domain, that also have experience designing software for the healthcare domain as well. They all have a unique understanding of questions that software practitioners ask about trace data, as they have all built tracing tools for software developers. The design activity included four sessions that took place on different days. The goal of the first two sessions was to gain an understanding of prior experience that our participants have had working with traceability, as well as to gather questions that they have about the trace prediction AI model. The last two sessions consisted of participatory design workshops set up to brainstorm ideas on how to represent the explanations.

Domain characterization. Based on feedback gathered during the interviews (the first two sessions), we identified the data domain to be *healthcare* (i.e., the artifacts are based on software designed for healthcare), the explanation domain to be *trace link generation*, and that the target users are *domain experts in software traceability.* These target users would mainly be interacting with trace link environments and therefore need a better understanding of why the software artifacts are predicted to be linked. Additionally, we identified and obtained raw data useful for answering the user questions through our discussion with our participants. We addressed the primary threat at this phase by interviewing our target users and gathering information that helped us identify questions that are relevant to their needs. After obtaining the final list of questions, we circled back with our users to ensure that the list was sufficient to capture all of the questions that they have about the AI model.

Explanation abstraction. Given the finalized collection of questions gathered from users, we analyzed and mapped these questions to abstract tasks (see Table 2). Further, based on the data determined to be relevant for answering these user questions, we abstracted these into data types, which we present in Table 3. To address the threats to validity at this phase, we met with our participants to verify that the set of abstract tasks aligned with their questions and that the data types are appropriate.

Explanation representations and interaction design. Through participatory design workshops which we conducted with our domain experts, we generated an array of design solutions for each of the abstract tasks. The role of each participatory design session was brainstorming different ideas for the explanations with the sole aim to generate *low-fidelity prototypes.* Researchers provided materials such as colored pencils, papers, and erasers, for each design activity. The lead researcher facilitated the design sessions by going through all of the user questions and their corresponding abstract tasks. The activity was then for all design participants (including researchers and the participants) to sketch on a piece of paper, how they would like to represent the explanation. We first provided a baseline representation for each of the abstract tasks to provide context (e.g., a matrix diagram for showing relationship). We then encouraged each participant (both the research participants and target user participants) to come up with as many design ideas as possible without focusing on aesthetics, but on the quantity of ideas. After each design activity, we asked each participant to present their sketch and discuss their thought process in coming up with the design. Researchers gathered notes and recordings for analysis. We selected the final representations for each abstract task based on the team's consensus on conditions such as how frequent the certain explanation representation appeared in the design solutions or how much the design solutions differed from a baseline representation.



The system shall improve accessibility of online clinical information and results.

The system shall provide the ability to receive and store general laboratory results (includes ability to differentiate preliminary results and final results and the ability to process a corrected result)

Results: Outcome of a the action carried out

results is a **parent_of** general laboratory results

Vetting Details

Date	Name	Role	Link correct?
11/06/19	Anon name 1	Software Engr	Yes
12/06/19	Anon name 2	Safety Analyst	Yes
11/06/19	Anon name 3	Software Engr	Yes
12/06/19	Anon name 4	Safety Analyst	Yes
11/06/19	Anon name 5	Software Engr	Yes
12/06/19	Anon name 6	Safety Analyst	No
11/06/19	Anon name 7	Requirements Engr	No

Figure 2: Overview of the software trace link prediction explanation interface. (A) and (B) shows the content of a source and target artifact, respectively. Color is used as a mapping between related concepts in the artifact content and their nodes in the relationship visualization graph. Size of each concept indicates its importance in the artifact content itself. (C) shows the relationship between the concepts in the source and target artifact. Hovering on a concept node in (C) would present a concept definition as shown in (E). Hovering on a link between concepts in (C) would present details about the relationship, as shown in (D). Users can see other experts that have vetted/unvetted a link (indicated by the thumbs up/down), as shown in (F). “Score” indicates the output predicted by the model on the semantic relationship between the two compared artifacts. "Type" indicates the artifact type; whether it is a requirement–R, source code–SC, or regulatory code–RC.

Based on user engagement and the co-design activities, we determined the interaction design between all of the explanation components and leveraged the Figma tool [9] for a medium-fidelity mock-up of the interaction design. Having a functional prototype allowed us to gather feedback from the target users on whether the interaction between each explanation component (e.g., linking, collapsing/expanding, hovering and updating) enables them to perform tasks for achieving their desired explanation goal. Based on the feedback gathered, we iterated through different designs for both the interaction aspect and for arranging the information on the interface screen. Once users were satisfied with how the prototype worked, we further implemented a web-based interface for the explanations (see Figure 2 for an overview of the interface). We validated the threat at this phase by consulting the literature on suitable design solutions for each of the abstract tasks. The participatory design sessions also allowed us to gather insights directly from our design partners.

5 CONCLUSIONS

With the aim to promote more human-centered explanation interface designs, we introduced a design decision framework to guide designers throughout the process of designing explanation interfaces. This framework consist of three phases, which ensures that designers (i) understand how an AI problem intersects with the user needs, (ii) consider user tasks and identify the relevant data for the explanations, and (iii) select appropriate representations and create the interaction design for the explanations in a coherent and useful way for the target user. We also discussed threats at each phase that can invalidate the design, as well as suggestions on how to address these threats. Further, we provided an example of how we applied our framework to design an explanation interface to help domain experts to better understand the prediction of links (traces) between software artifacts in the healthcare domain. Outcomes of our work demonstrate the need to approach the design of explanation interfaces from a human-centered perspective, putting users at the core of the design process. In the future, we plan to evaluate the finalized design with expert users that are different from those that participated in the design process. We also plan to investigate the temporal (dialogue) aspect of an explanation and how to operationalize it for AI explanations.

6 ACKNOWLEDGEMENT

This work has been supported by the NSF award 1901059. We also appreciate the software engineering experts that participated in our co-design study.

REFERENCES

[1] David Alvarez-Melis and Tommi S Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. *arXiv preprint arXiv:1707.01943* (2017).

[2] Dustin L Arendt, Nasheen Nur, Zhuanyi Huang, Gabriel Fair, and Wenwen Dou. 2020. Parallel embeddings: a visualization technique for contrasting learned representations. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 259–274.

[3] Ángel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. 2019. Fairvis: Visual analytics for discovering intersectional bias in machine learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 46–56.

[4] Bei Chen, Rahul Nair, and Inge Vejsbjerg. 2019. RSM: An explainable predictive sales route selector. In *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 1090–1093.

[5] Hanjie Chen and Yangfeng Ji. 2020. Learning Variational Word Masks to Improve the Interpretability of Neural Text Classifiers. *arXiv preprint arXiv:2010.00667* (2020).

[6] Eric Chu, Deb Roy, and Jacob Andreas. 2020. Are visual explanations useful? a case study in model-in-the-loop prediction. *arXiv preprint arXiv:2007.12248* (2020).

[7] Danilo Croce, Daniele Rossini, and Roberto Basili. 2019. Auditing deep learning processes through kernel-based explanatory models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 4028–4037.

[8] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. *arXiv preprint arXiv:2010.00711* (2020).

[9] Figma Design. 2017. Figma: the collaborative interface design tool.(2017). *Retrieved September* 17 (2017), 2017.

[10] Upol Ehsan, Q Vera Liao, Michael Muller, Mark O Riedl, and Justin D Weisz. 2021. Expanding Explainability: Towards Social Transparency in AI systems. *arXiv preprint arXiv:2101.04719* (2021).

[11] Sebastian Gehrmann, Hendrik Strobelt, Robert Krüger, Hanspeter Pfister, and Alexander M Rush. 2019. Visual interaction with deep learning models through collaborative semantic inference. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 884–894.

[12] John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4129–4138.

[13] Fred Hohman, Arjun Srinivasan, and Steven M Drucker. 2019. TeleGam: Combining visualization and verbalization for interpretable machine learning. In *2019 IEEE Visualization Conference (VIS)*. IEEE, 151–155.

[14] Vijay Kakani, Van Huan Nguyen, Basivi Praveen Kumar, Hakil Kim, and Visweswara Rao Pasupuleti. 2020. A critical review on computer vision and artificial intelligence in food industry. *Journal of Agriculture and Food Research* 2 (2020), 100033.

[15] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 5686–5697.

[16] Simon Meyer Lauritsen, Mads Kristensen, Mathias Vassard Olsen, Morten Skaarup Larsen, Katrine Meyer Lauritsen, Marianne Johansson Jørgensen, Jeppe Lange, and Bo Thiesson. 2020. Explainable artificial intelligence model to predict acute critical illness from electronic health records. *Nature communications* 11, 1 (2020), 1–11.

[17] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. 2015. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics* 9, 3 (2015), 1350–1371.

[18] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Explainable outfit recommendation with joint outfit matching and comment generation. *IEEE Transactions on Knowledge and Data Engineering* 32, 8 (2019), 1502–1516.

[19] Tim Miller, Piers Howe, and Liz Sonenberg. 2017. Explainable AI: Beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioural sciences. *arXiv preprint arXiv:1712.00547* (2017).

[20] Sina Mohseni, Niloofar Zarei, and Eric D Ragan. 2018. A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems. *arXiv* (2018), arXiv–1811.

[21] Henrik Mucha, Sebastian Robert, Ruediger Breitschwerdt, and Michael Fellmann. 2021. Interfaces for Explanations in Human-AI Interaction: Proposing a Design Evaluation Approach. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–6.

[22] Henrik Mucha and Franziska Schulz. 2021. Explanation Interfaces: Two Approaches for Grounding Design Decisions. *ACM CHI Workshop on Operationalizing Human-Centered Perspectives in Explainable AI* (2021). https://hcxai.jimdosite.com/papers/

[23] Tamara Munzner. 2009. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics* 15, 6 (2009), 921–928.

[24] Jarmila Novotná. 2005. Teachers' views and use of explanation in teaching mathematics. In *international symposium on elementary mathematics teaching, Prague, Czech Republic*.

[25] Osonde A Osoba and William Welser IV. 2017. *An intelligence in our image: The risks of bias and errors in artificial intelligence*. Rand Corporation.

[26] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Vol. 13-17-August-2016. Association for Computing Machinery, 1135–1144. https://doi.org/10.1145/2939672.2939778 arXiv:1602.04938

[27] Edward Segel and Jeffrey Heer. 2010. Narrative visualization: Telling stories with data. *IEEE transactions on visualization and computer graphics* 16, 6 (2010), 1139–1148.

[28] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[29] Hua Shen and Ting-Hao Huang. 2020. How Useful Are the Machine-Generated Interpretations to General Users? A Human Evaluation on Guessing the Incorrectly Predicted Labels. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 8. 168–172.

[30] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. 2019. Designing theory-driven user-centric explainable AI. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–15.

[31] Gary KW Wong, Xiaojuan Ma, Pierre Dillenbourg, and John Huan. 2020. Broadening artificial intelligence education in K-12: where to start? *ACM Inroads* 11, 1 (2020), 20–29.

[32] Fan Yang, Shiva K Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D Ragan, Shuiwang Ji, and Xia Hu. 2019. XFake: explainable fake news detector with visualizations. In *The World Wide Web Conference*. 3600–3604.

[33] Qiyu Zhi, Alvitta Ottley, and Ronald Metoyer. 2019. Linking and layout: Exploring the integration of text and visualization in storytelling. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 675–685.

[34] Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. 2018. An interpretable reasoning network for multi-relation question answering. *arXiv preprint arXiv:1801.04726* (2018).