Explainable AI for Constraint-Based Expert Systems

Jan Bode
Karlsruhe Institute of Technology, jan.bode@student.kit.edu

Max Schemmer
Karlsruhe Institute of Technology, max.schemmer@kit.edu

Tomáš Balyo
Future Labs, CAS Software, tomas.balyo@cas.de

Explainable AI for Constraint-Based Expert Systems

Jan Bode[1,2], Max Schemmer[1], and Tomáš Balyo[2]

[1] KIT, Karlsruhe, Germany
jan.bode@student.kit.edu, max.schemmer@kit.edu
[2] Future Labs, CAS Software, Karlsruhe, Germany
tomas.balyo@cas.de

Abstract. *The need to derive explanations from machine learning (ML)-based AI systems has been addressed in recent research due to the opaqueness of their processing. However, a significant amount of productive AI systems are not based on ML but are expert systems including strong opaqueness. A resulting lack of understanding causes massive inefficiencies in business processes that involve opaque expert systems. This work uses recent research interest in explainable AI (XAI) to generate knowledge for the design of explanations in constraint-based expert systems. Following the Design Science Research paradigm, we develop design requirements and design principles. Subsequently, we design an artifact and evaluate the artifact in two experiments. We observe the following phenomena. First, global explanations in a textual format were well-received. Second, abstract local explanations improved comprehensibility. Third, contrastive explanations successfully assisted in the resolution of contradictions. Finally, a local tree-based explanation was perceived as challenging to understand.*
Keywords: Explainable Artificial Intelligence, Expert Systems, Constraints, Configuration

1 Introduction

Expert systems are one of the most successful and widespread applications of AI [1–3]. They are designed to emulate the decision-making process of human experts [4] and find applications in various industries, including engineering, telecommunication, smart manufacturing, and construction [3, 5, 6].

While traditional expert systems rely on rule-based knowledge representation in form of if-then rules [7], more modern systems dependent on complex and opaque models [8, 9]. A particularly important type of model-based expert systems are constraint-based approaches because they offer great functionality in terms of knowledge representation and inferential competence due to a separation of knowledge representation and reasoning [10, 11]. Instead of if-then rules, they depend on constraints to formulate relationships between variables and constraint solvers to compute a valid solution. Nevertheless, constraint-based expert systems do not come without their challenges. While the reasoning of constraint-based expert systems is more efficient compared to rule-based systems [9], it relies on *black box* constraint solvers that compute an output based on an input in form of constraints [12]. These advanced reasoning capabilities confront users with the problem that certain decisions are increasingly hard to understand [13].

17th International Conference on Wirtschaftsinformatik
February 2022, Nürnberg, Germany

Experience in business has shown that a lack of understanding leads to huge inefficiencies in the knowledge base debugging process. As a result, businesses rely on highly skilled developers to resolve the errors and comprehensibility issues. However, this is resource and time-consuming for all involved stakeholders. Therefore, constraint-based expert systems, should provide explanations on their reasoning.

Explainable AI (XAI) is a subfield of AI with the goal to make AI more understandable given a specific desideratum [14]. While XAI is not new, it has seen a renewed public and academic interest due to the rise of increasingly opaque machine learning (ML)-based systems that rely on black-box algorithms [15]. Most of the recent publications in XAI express an algorithm-centric view and rely on the researcher's intuition of a good explanation [16]. This is problematic because explanations for AI systems are usually requested by stakeholders with low technical understanding [17]. Therefore, an interdisciplinary collaboration between the fields of AI, social sciences, and human-computer-interaction is needed [16]. Furthermore, current approaches lack commonly agreed-on design principles for developing user-centric XAI systems [17].

In this work, we use a user-centric approach that combines knowledge from recent publications in XAI and social sciences to derive design knowledge for explainable constraint-based expert systems and thereby answer the following research question:

RQ: *How to design explanations in constraint-based expert systems?*

This paper aims to answer the research question by performing a full Design Science Research (DSR) cycle consisting of the following phases: *Awareness of the problem*, *Suggestions*, *Development*, *Evaluation* and *Conclusion* [18]. We derive, implement, and evaluate design requirements and design principles for user-centric explanations in constraint-based expert systems. We identify design issues users currently face in two steps. First, we identify issues in a review of recent publications. Second, we conduct expert interviews with multiple stakeholders of a constraint-based product configurator. Product configurators serve as a great example because they have a high level of human-computer interaction. We use the identified issues to derive design requirements and design principles. In the next step, we use the derived design principles to develop an artifact. The artifact consists of an explanation subsystem, which we integrate into an existing software product. For the evaluation of the DSR project, we conduct two think-aloud experiments and a follow-up interview with two participants to test the utility of the explanation subsystem. The result of this work is design knowledge in the form of design principles and design requirements. Additionally, we point out implications for future research in XAI.

2 Foundations & Related Work

In this Section we describe the theoretical foundations for the design of explanation subsystems in constraint-based expert systems and introduce related work in this field.

2.1 Expert Systems

Expert systems are a subfield of AI [19] and are computer programs that use or represent the knowledge of human experts to provide high-quality performance in a specific domain [3]. Expert systems are used to automate and guide decision-making and problem-solving processes [3]. The most widespread types of expert systems are rule-based and model-based systems. Rule-based systems rely on *if-then* rules to represent knowledge. For example, *if* temperature below 0°C, *then* water freezes. Model-based systems rely on a model to represent complex relationships between variables. The main advantage in comparison with rule-based systems is the separation of knowledge representation and reasoning [11]. *Constraint-based systems* are a specific type of model-based expert systems. They rely on constraints to model relationships between variables [20]. Constraints restrict the number of possible solutions. For a valid solution, all constraints must be satisfied. This approach has two main strengths. First, it allows users to reach new innovative solutions. Second, it can prove that problems have no solution. The drawback is the work associated with the correct modeling of the domain knowledge. For example, a car manufacturer reserves specific colors for the premium model. This sets a constraint on the available color options for the base model.

2.2 Explainable Artificial Intelligence (XAI)

XAI is concerned with the design of explanations that make AI systems understandable for human stakeholders [16]. Explanations are required if AI systems are too complex or too opaque to allow for human oversight [21]. However, while many researchers agree on the goal of *making AI systems understandable*, different stakeholders have different desiderata. The term desiderata refers to stakeholder's "interests, goals, expectations, needs, and demands regarding AI systems" [14, 2]. We differentiate between three categories of explanations: *global, local*, and *counterfactual*.

Global explanations. Global explanations aim to support the general understanding of the AI system. They do not provide information on a certain output or decision. Global methods aim to answer *how-questions* [17]. For example, *how does the system work?*. They provide general information. Common global explanation methods are: *global feature importance, decision tree approximation*, and *rule extraction* [17]. Global feature importance describes the influence of specific features for the whole model. Decision trees approximate and visualize possible decision paths, while rule-extraction approximates the model with a set of rules.

Local explanations. Local explanations aim to support the understanding of a specific output or decision the system reached. They refer to *why-questions* [17]. Local explanations provide insights, *why a system reached a specific conclusion*. Common methods are *local rules*, and *local trees*. Local rules describe the rules used to reach a certain conclusion. Local trees visualize the decision tree path.

Counterfactual explanations. Counterfactual explanations inspect how changes in the input may influence the output. They allow users to assess how they can reach a different (desired) decision. They refer to *what if, why, why not, and how to be that-questions* [17]. For example, *how can I reach a more favorable insurance class? Contrastive features* is a common method that describes which feature needs to be changed for the system to reach a different (desired) conclusion or output.

2.3 Related Work

Recent publications include the proposal of a framework for explanation subsystems in rule-based systems and step-wise explanations of constraint satisfaction problems [13,22]. The framework proposes a combination of global- and local explanations in natural language [22]. However, the focus on rule-based expert systems limits the transferability to this work. The step-wise approach calculates and visualizes each reasoning step in a matrix to make the reasoning more transparent and understandable [13]. As a result, the approach is promising for explanations with a high need for detail. However, the explanation steps drastically increase with more complex problems. This makes the step-wise approach unfeasible for complex problems.

3 Methodology

In this paper we utilize the methodology of DSR. DSR "is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence" [23, 5]. The goal of DSR is to generate design knowledge and theoretical insights [24]. This can be achieved by building a theory-based artifact, or by implementing empirically derived design principles [24]. This work focuses on the latter.

Following Vaishnavi and Kuechler (2015), a DSR cycle is an iterative process consisting of five phases, namely: *Awareness of the problem*, *Suggestions*, *Development*, *Evaluation* and *Conclusion* [18]. This work completes one iteration of the DSR cycle.

As a part of the DSR cycle we design, develop, and evaluate an explanation subsystem for the constraint-based product configurator *Merlin*[1]. Configuration is a design activity of an artifact, based on a set of predefined components [25]. Configuration plays a leading role in the paradigm shift from mass production to mass customization [26].

4 Designing an Explanation Subsystem

This Section describes the design process of an explanation subsystem for the constraint-based product configurator Merlin. It is structured according to the DSR phases presented in Section 3.

4.1 Awareness of Problem

In the *awareness of problem* phase, we conduct semi-structured expert interviews to define the domain problem and analyze stakeholder desiderata. Additionally, we supplement the user desiderata with a review of recent publications in XAI.

[1] Merlin is a commercial product configurator developed by CAS Software `https://www.cas-merlin.de/`. It consists of three main components: M.Sales, M.Core, and M.Model. *M.Sales* facilitates the configuration activity. *M.Core* is responsible for the reasoning and *M.Model* allows users to access and edit the knowledge base.

We select semi-structured expert interviews because they provide in-depth insights and allow for open-ended and follow-up questions [27]. The goal of the expert interviews is to precisely identify the problem and understand users. We conducted the interviews with ten CAS employees from different divisions. In Table 1, we provide demographic data for all interviewees. We analyze the interviews using template analysis in MAXQDA [28]. We paraphrase all paragraphs and subsequently assign them to codes. We repeat this process for a second and third iteration to refine the code system. The interviews helped to identify the domain problem. During the interactive configuration task users can run into *contradictions*. Contradictions occur if a desired configuration is not available due to a set of constraints in the model. End-users face the problem that the configurator does not provide helpful feedback in case of a contradiction. As a result, users have a low understanding of the problem. This frequently leads to faulty or imprecise error reports that cause an inefficient and costly support workflow.

We derive the following user issue during the interviews: users are *unable to comprehend the contradiction* (Alpha, Iota). This results in several sub-issues. First, a *lack of transparency* (Epsilon, Eta). Second, users are *unable to verify the correct reasoning* (Kappa). Third, users are *unable to directly report an error* (Delta, Iota, Kappa) and last, users *do not receive recommendations on how to resolve the contradiction* (Alpha).

In addition we derive the following desiderata from recent publications: *trust, usability, usefulness, verification* [14].

Table 1. Interviewee demographics

Interviewee	α	β	γ	δ	ϵ	ζ	η	θ	ι	κ	ø
Age	22	19	25	26	26	26	35	49	30	25	28.3
Sex	M	M	M	F	M	M	M	M	M	M	/
Duration (min.)	23	25	25	29	31	25	24	45	46	51	32.4

4.2 Suggestions

In this phase, we develop design requirements based on issues and literature desiderata identified in the previous phase. Afterward, we derive design principles based on the design requirements and recent publications.

Design Requirements (DR). The most frequently stated issue is that users do not comprehend the contradiction (Alpha, Iota). They do not understand *why* the contradiction occurs, neither do they understand *how* the contradiction occurs. Both question types refer to different types of explanation. The following requirement addresses the *how-question*.

DR 1: *The system should provide insights on how the general reasoning process works.*

"It would be most helpful to know why, why does a contradiction occur?" (Gamma). We address the *why-question* with the provision of selected information. The term *selected* is very important for this requirement. It addresses the issue that presenting the cause attribution not necessarily qualifies as an explanation [16]. This is because people do not expect the complete cause, but rather a *selected* subset of causes [16].

DR 2: *The system should provide selected information on why it made a certain decision.*

"First of all, you need to able to trust the system, you need to be able to verify a decision" (Eta). Another derived issue is the lack of transparency. A related literature desideratum is trust, since improved transparency fosters trust in AI systems [29]. Based on this, we derive the following requirement for explanation subsystems.

DR 3: *The system should be transparent regarding its capabilities and limitations.*

It would be great to get some kind of recommendation on how to resolve the contradiction (Alpha). During the interviews, we identified the issue that users cannot verify the reasoning process and are unable to assist in the resolution of a contradiction.

DR 4: *The system should assist users in the verification of the reasoning process and the resolution of the contradiction.*

Finally, we address the desideratum of usability. A more usable system can help users to reach a decision more quickly, and improve the decision quality [30].

DR 5: *The system should be easy and intuitive to use.*

Design Principles (DP). This work uses the conceptual schema proposed in [31] to formulate design principles based on the identified user-requirements. The following applies to all design principles. Implementers are designers and developers. Recipient users are end-users of the product configurator. The aim and context of the design principles correspond with the aim of this paper: *The design of user-centric explanations in constraint-based expert systems*.

The first design principle aims to satisfy requirements 1 and 3. Requirement 1 states that the system should provide information on *how* it works. Therefore, we propose to provide global explanations [17]. Furthermore, we address the issue of transparency formulated in Requirement 3.

DP 1: *Systems should provide a global explanation to increase transparency, trust, and global comprehensibility.*

The provision of a global explanation does not conflict with the provision of a local explanation. In contrast, local and global approaches can serve to reinforce one another [32]. The following design principle addresses Requirement 2 and 3. The *why-question* requires a local explanation [17], while *information* in constraint-based expert systems refers to features, characteristics, and variables and their relationships. This measure also increases transparency by ensuring inspectability [29].

DP 2: *Systems should provide a local explanation containing selected information on features, characteristics, and variables and their relationships to improve comprehensibility and transparency.*

The third design principle addresses Requirement 4. Requirement 4 states that the systems should assist users in the resolution of the contradiction. In our case, this refers to the *how to be that* question type. This question type requires a contrastive explanation [16, 17].

DP 3: *Systems should provide a contrastive explanation to assist in the resolution of the contradiction.*

The following design principle address Requirement 5 - *the system should be easy and intuitive to use*. In recent publications, many approaches, guidelines, and frameworks exist on how to design user-centric explanations, i.a. [33]. However, we address the requirement with a reduction of complexity - a usability issue stated during the interviews. We achieve this with the provision of information at different levels of abstraction. We derive the design principle as follows.

DP 4: *Systems should provide information at different levels of abstraction and detail to reduce complexity and improve comprehensibility and usability.*

We provide an overview of the design principles and addressed design requirements in Table 2.

Table 2. Design principles and addressed design requirements

Design Principle	Description	Addressed Requirements
DP 1	Systems should provide a global explanation to increase transparency, trust, and global comprehensibility.	1, 3
DP 2	Systems should provide a local explanation containing selcted information on features, characteristics, and variables and their relationships to improve comprehensibility and transparency.	2, 3
DP 3	Systems should provide a contrastive explanation to assist in the resolution of the contradiction.	4
DP 4	Systems should provide information at different levels of abstraction and detail to reduce complexity and improve comprehensibility and usability.	5

4.3 Development

The explanation subsystem itself is embedded in the product configurator Merlin. Since the subsystem is directed towards end-users, we embedded it in the configuration interface of Merlin – M.Sales. In our case, the explanation subsystem aims to improve the understanding of contradictions that occur during the interactive configuration process. Therefore, the subsystem provides an explanation if a configuration step caused a contradiction. Based on the design principles, we derive the following design features.

Two-step Explanation. To address *DP 4*, we provide the explanation in two steps. At the first step the user receives a notification that a contradiction occurred. The notification provides basic information about the contradiction and allows the user to request a *Detailed report*. Figure 1 and 2 visualize both steps for a specific contradiction. The detailed report provides the following types of explanations.

Global Explanation. We provide an abstract global explanation to address *DP 1*, and *DP 4*. A predefined text explains the user *how* the system reasons.

Local Explanation. Furthermore, we divide the local explanation into two parts to address *DP 2*, and *DP 4*. We provide abstract information about the involved components

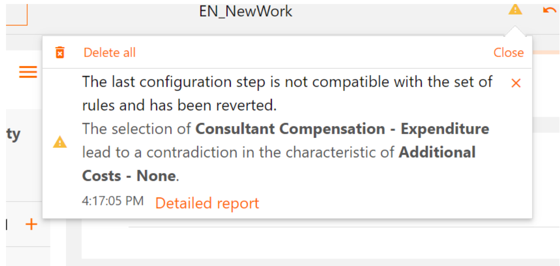

Figure 1. Developed explanation subsystem: Notification

of the contradiction and a an interactive tree-based explanation. The *Implication Tree* provides further information about involved components and their relationships.

Recommendation. Finally, we provide a recommendation on how to resolve the contradiction to address *DP 3*, and *DP 4*.

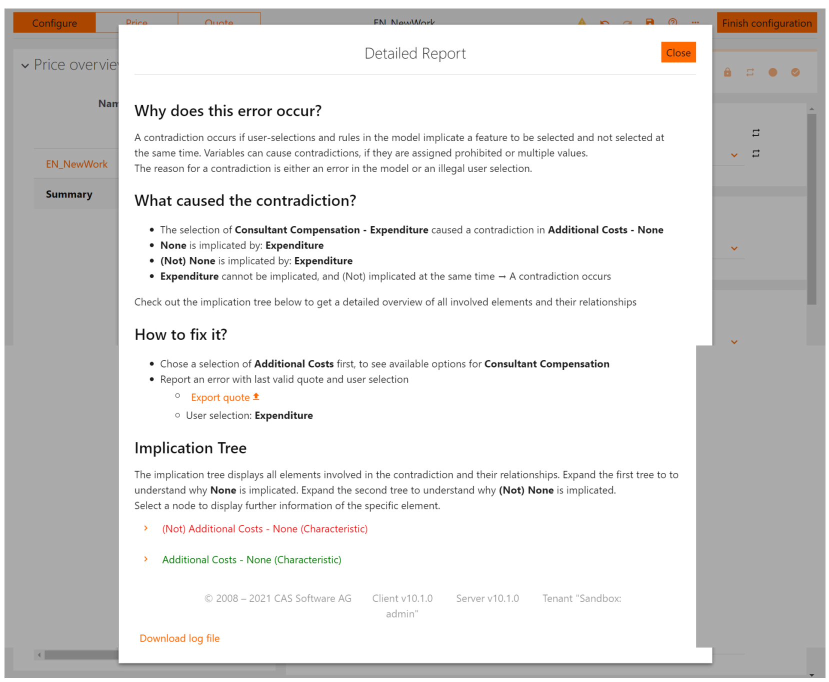

Figure 2. Developed explanation subsystem: Detailed report

4.4 Evaluation

To test the utility of the artifact, we conduct two separate interviews via video call with screen sharing. The goal of the evaluation is to test whether the explanation subsystem provides additional utility for the system. The evaluation consists of two parts. In the first part, we conduct an experiment. To assess the utility, we confront the evaluands with a configuration task and a set of instructions on how to respond to contradictions. In the second part, we ask the participants to provide additional feedback. We did not use the problem instances for the development or testing of the artifact. They present real-world problems users of the product configurator currently face. However, they are modified for privacy reasons and to fit the scope of this work. We conduct the experiment with two employees from CAS. Both participants have experience in the use of the end-user interface. For better comparability only participant *Mu* had access to the explanation subsystem for the first part of the evaluation. Participants *Mu*, and *Lambda* had access to the explanation subsystem for the second part of the evaluation.

For the first experiment, the participants have to configure a product according to a predefined set of attributes. The configuration task is designed to trigger a contradiction. Among others, the participants have to assign the value *4* to the variable *Lead Time*. However, the *Type of Product - Single Contract* implies values of five or higher for *Lead Time*. The characteristic *Type of Product* is a default value and cannot be changed. Therefore, users get a contradiction if they try to assign the value *4* to *Lead Time*. Figure 3 visualizes the rule responsible for the contradiction. In case of a contradiction, the participants have to report an error and modify the configuration to resolve the contradiction. The error report examines whether the participants understand the problem at hand. The resolution of the contradiction assesses the ability of the participants to resolve the contradiction based on the available information. Since users are unable to modify the rules, they have to modify the configuration. However, their goal is to reduce the deviations from the predefined configuration task to a minimum.

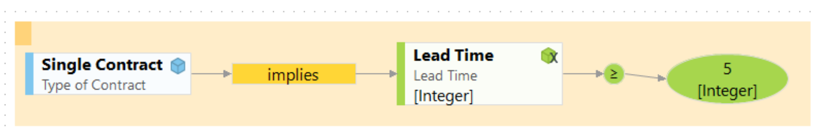

Figure 3. Rule responsible for contradiction

In the second part of the evaluation, we ask the participants to provide additional feedback on different components of the explanation subsystem. For this purpose, we present the explanation subsystem to both participants.

Results. In the experiment, we assess how well the participants understand the problem. During the configuration task, the participants triggered a contradiction. The first observation we made is the formulation of the problem description. Both participants used the same template for the error report. They had to fill in the name of the product, the configuration step that caused the contradiction, the expected behavior, and the last valid configuration. The error report the participants provided only differed in *expected behavior*. Mu expected a specific variable value, while Lambda expected *no error*. This indicates a better understanding of the problem for Mu. In the second task, the participants had to generate a similar but valid offer. The part of the experiment disclosed the problem discussed in *awareness of problem*. End-users do not receive feedback on *why a contradiction occurs*. Lambda tried several options to create an offer but failed. In our experiment, this led to dissatisfaction and consumed a lot of time. Mu, however, had a different approach and used information from the explanation subsystem to identify the problem and create an offer. While this approach was more successful, it was also relatively time-consuming. Mu stated that parts of the explanation, to be precise the implication tree, was not intuitive neither easy to understand. This was due to the complex structure of the implication tree. Feedback from the second part of the evaluation further indicates a comprehensibility problem with the tree-based explanation.

We infer that the implementation of the design principles indicates an improved end-user problem-understanding. However, the weaknesses in the implication tree point out a need for improvement of the implication of design principle 4 - the presentation of information at different levels of abstraction and detail to reduce complexity and improve comprehensibility.

5 Discussion

In this Section we discuss the results of this paper. Furthermore, we highlight the limitations and contributions and present implications for future work.

The goal of this paper is to derive design knowledge for user-centric explanations in constraint-based expert systems. To answer the research question, we analyze user desiderata and issues to derive design principles. We derive desiderata from recent publications and formulate issues based on a series of expert interviews. This work shows that users tend to distrust artificial systems due to low transparency and low comprehensibility of the overall system. We propose a global explanation to address the described issues. The global explanation allows the users to understand how the system work and therefore builds trust and (global) transparency. The global explanation was also well received in the evaluation of the artifact.

Furthermore, we found that users have difficulties comprehending the system's reasoning in case of a contradiction. They do not understand *why* a contradiction occurs. To address this issue, we provide a local explanation that provides the user with selected information on the contradiction. Thereby, we improve transparency and comprehensibility of specific reasoning steps.

Moreover, we identify the issue that users need some form of guidance or recommendation to resolve upcoming contradictions. Therefore, we propose that systems should assist the user in the resolution with the provision of a contrastive explanation. This principle was well-received during the evaluation and helped the user to resolve a contradiction.

Additionally, we identify the problem that users have difficulties comprehending complex contradictions. However, at the same time, they require detailed information. We propose the general provision of information relevant for the explanation subsystem at different levels of abstraction and detail. This allows the explanation subsystem to provide detailed information while maintaining comprehensibility.

5.1 Contributions

In addition to the explanation subsystem itself, we derived design principles that serve as guidelines for future design and development processes of explanation subsystems. The evaluation indicates that the implementation of the design principles improves the end-user's understanding of contradictions. We acknowledge that none of the design principles propose any new explainability approaches. However, to our knowledge, the proposal of the combination of global, local, and contrastive explanations at various levels of abstraction and detail is a new and promising design approach.

5.2 Limitations

In this Section, we describe the limitations of this work and discuss the generality of the results. We structure the order of the limitations according to the DSR phases. In *awareness of the problem* we conducted ten expert interviews which is rather small. Furthermore, none of the interviewees were *real* end-users of the Configurator Merlin. To assure that all problems are identified correctly, we want to conduct further interviews that include real end-users in the next DSR iteration. However, we argue that we can approximate user needs based on the diverse domain knowledge of all experts. Furthermore, we noticed that few problems were repeatedly stated. Therefore, we cannot say with certainty how much additional knowledge will be gained in future interviews.

During the *development* phase, we used different problem instances that can trigger a contradiction. However, each new problem instance posed new challenges and sometimes bugs. Therefore, we need to test the developed explanation subsystem with more real problem instances to guarantee a bug-free system.

Another problem is that we conducted the *evaluation* with two participants. This sample size is too small to give statistically valid statements. Therefore, the results of the evaluation can only function as an indicator for the next DSR iteration. Furthermore, we conducted the evaluation via an online video call using screen sharing. This has the limitations that we could not guarantee the same environment for the experiment, e.g., different internet speeds or a second monitor. Additionally, the evaluands were employees from CAS. Both had to solve an end-user problem instance. But none of the evaluands is an end-user, e.g., salesperson. However, both evaluands have experience in the use of the product configurator. Therefore, we argue that they are still a viable choice. However, for future work, we want to thoroughly evaluate the artifact with *real* end-users.

Another limitation we identified during the evaluation is that the evaluands perceived the implication tree as *unintuitive* and *hard to understand*. Therefore, future work should assess the implementation of design principles 2 and 4 - *the provision of an interactive local explanation* - and the design principles themselves.

5.3 Future Work

The evaluation showed that participants had difficulties understanding the interactive local explanation. Therefore, future work should focus on the design of detailed interactive local explanations. Our work showed that it is especially hard to design a detailed explanation that is still comprehensible. Research suggest that explanations should be *selected* out of a number of causes [16]. However, it is very hard to select the right amount of causes due to different levels of expertise. We suggest that future work should assess new approaches on how to present detailed causal information in a comprehensible way. An interesting approach could be to use step-wise explanations as proposed by [13]. The process could be further enhanced by assessing how users use step-wise explanations to *select* or focus on specific explanation steps.

Another issue we faced is that the expert system cannot verify the reasoning steps. Future work should therefore assess different verification options. One way would be to collect data during each configuration process. This could guide end-users in case of an incorrect configuration step and give specific recommendations based on previous configuration tasks. A different approach would be to train a machine learning model based on labeled data. Data could be generated using interactive labeling during the configuration process.

6 Conclusion

The lack of understanding for reasoning processes in constraint-based expert systems causes huge inefficiencies in business processes. However, current research focuses on ML-based systems and neglect the importance of the social sciences and user-centricity. Furthermore, there is a lack of commonly agreed-on design principles.

To address this issue, we derived design knowledge for user-centric explanations in constraint-based expert systems. We address the research problem with a full iteration of the DSR paradigm. In the first phase, we derived desiderata and issues for expert interviews and a literature review. The most significant issue for users is the lack of comprehensibility. Based on design requirements we derived from the issues and desiderata, we derived four design principles. Design principles 1-3 state that explanation subsystems should provide a global, local, and contrastive explanation. The explanations themselves should provide selected information on different levels of detail and abstraction (design principle 4).

We developed an artifact to implement and evaluate the design principles and requirements. We evaluated the artifact and, therefore, the design principles in two steps. First, we conducted an experiment to test the utility of the artifact for end-users. Second, we asked the participants to provide additional feedback for the artifact. The evaluation indicates that the artifact increases the overall utility of the system. However, the participants perceived the interactive tree-based explanation as too complex and unintuitive. As a result, we need to reassess the related design principles and explore different methods for local explanations in the next DSR iteration. Furthermore, future work could assess the suitability of tree-based explanations in the field of XAI.

References

1. Yoon, Y., Guimaraes, T., O'Neal, Q.: Exploring the factors associated with expert systems success. MIS Quarterly 19(1), 83–106 (1995)
2. Zhang, L., Pan, Y., Wu, X., Skibniewski, M.J.: Expert Systems, pp. 201–230. Springer Singapore, Singapore (2021)
3. Shang, Y.: 5 - expert systems. In: CHEN, W.K. (ed.) The Electrical Engineering Handbook, pp. 367–377. Academic Press, Burlington (2005)
4. Jackson, P.: Introduction to Expert Systems. Addison-Wesley Longman Publishing Co., Inc., USA, 3rd edn. (1998)
5. Adamczyk, B.S., Szejka, A.L., Canciglieri, O.: Knowledge-based expert system to support the semantic interoperability in smart manufacturing. Computers in Industry 115, 103161 (2020)
6. Amiri, M., Ardeshir, A., Fazel Zarandi, M.H.: Fuzzy probabilistic expert system for occupational hazard assessment in construction. Safety Science 93, 16–28 (2017)
7. Bowen, J., Likitvivatanavong, C.: Question-generation in constraint-based expert systems (2002)
8. Salem, H., Soria, D., Lund, J.N., Awwad, A.: A systematic review of the applications of Expert Systems (ES) and machine learning (ML) in clinical urology. BMC Medical Informatics and Decision Making 21(1), 223 (Jul 2021)
9. Neumann, M., Baumeister, J.: A rule-based vs. a model-based implementation of the knowledge system limpact and its significance for maintenance and discovery of ecological knowledge (2002)
10. Bowen, J.: Constraint Processing Offers Improved Expressiveness and Inference for Interactive Expert Systems. In: O'Sullivan, B. (ed.) Recent Advances in Constraints. pp. 93–108. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2003)
11. Friedrich, G., Jannach, D., Stumptner, M., Zanker, M.: Chapter 11 - Knowledge Engineering for Configuration Systems. In: Felfernig, A., Hotz, L., Bagley, C., Tiihonen, J. (eds.) Knowledge-Based Configuration, pp. 139–155. Morgan Kaufmann, Boston (Jan 2014)
12. Runte, W.: Enhancing business process management with a constraint-based approach. In: SoMeT (2012)
13. Bogaerts, B., Gamba, E., Claes, J., Guns, T.: Step-wise explanations of constraint satisfaction problems. In: 24th European Conference on Artificial Intelligence (ECAI). IOS Press, Netherlands (Jan 2020)
14. Langer, M., Oster, D., Speith, T., Hermanns, H., Kästner, L., Schmidt, E., Sesing, A., Baum, K.: What do we want from Explainable Artificial Intelligence (XAI)? - A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research. Artificial Intelligence 296, 103473 (Jan 2021)
15. Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58, 82–115 (Jun 2020)
16. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences (2018)
17. Liao, Q.V., Gruen, D.M., Miller, S.: Questioning the AI: informing design practices for explainable AI user experiences. CoRR abs/2001.02478 (2020)
18. Vaishnavi, V.K., Kuechler, W.: Design Science Research Methods and Patterns. CRC Press (Jan 2015)
19. Chowdhary, K.R.: Fundamentals of artificial intelligence. Springer India, New Delhi (Jan 2020)
20. Flannery, L.M., Gonzalez, A.J.: Detecting anomalies in constraint-based systems. Engineering Applications of Artificial Intelligence 10(3), 257–268 (Jun 1997)

21. Burrell, J.: How the machine ‘thinks’: Understanding opacity in machine learning algorithms. Big Data and Society 3(1) (2016)
22. Tomic, B., Horvat, B., Jovanovic, N.: An explanation facility framework for rule-based systems. International Journal on Artificial Intelligence Tools 21 (2012)
23. Hevner, A., Chatterjee, S.: Design Research in Information Systems: Theory and Practice. Integrated Series in Information Systems, Springer US (2010)
24. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
25. Mittal, S., Frayman, F.: Towards a generic model of configuraton tasks. In: Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 2. p. 1395–1401. IJCAI’89, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)
26. Sabin, D., Weigel, R.: Product configuration frameworks-a survey. IEEE Intelligent Systems and their Applications 13(4), 42–49 (1998)
27. Adams, W.C.: Conducting Semi-Structured Interviews, chap. 19, pp. 492–505. John Wiley & Sons, Ltd (2015)
28. VERBI Software: MAXQDA 2020 [computer software]. Berlin, Germany: VERBI Software. Available from maxqda.com. (2019), `maxqda.com`
29. Felzmann, H., Fosch-Villaronga, E., Lutz, C., Tamo-Larrieux, A.: Towards Transparency by Design for Artificial Intelligence. Science and Engineering Ethics 26(6), 3333–3361 (Dec 2020)
30. Endsley, M.R.: From here to autonomy: Lessons learned from human–automation research. Human Factors 59(1), 5–27 (2017), pMID: 28146676
31. Gregor, S., Kruse, L., Seidel, S.: Research Perspectives: The Anatomy of a Design Principle. Journal of the Association for Information Systems 21(6), 1622–1652 (Jan 2020)
32. Mueller, S.T., Veinott, E.S., Hoffman, R.R., Klein, G., Alam, L., Mamun, T.I., Clancey, W.J.: Principles of explanation in human-ai systems. ArXiv abs/2102.04972 (2021)
33. Wang, D., Yang, Q., Abdul, A., Lim, B.Y.: Designing Theory-Driven User-Centric Explainable AI. In: Brewster, S., Fitzpatrick, G., Cox, A., Kostakos, V. (eds.) Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–15. ACM, New York, NY, USA (2019), num Pages: 15