From Facts to Foils: Designing and Evaluating Counterfactual Explanations for Smart Environments

Anna Trapp
University of Cologne
Cologne, Germany
atrapp2@smail.uni-koeln.de

Mersedeh Sadeghi*
University of Cologne
Cologne, Germany
mersedeh.sadeghi@uni-koeln.de

Andreas Vogelsang
paluno - The Ruhr Institute for Software Technology
University of Duisburg-Essen, Essen, Germany
andreas.vogelsang@uni-due.de

***Abstract*—Explainability is increasingly seen as an essential feature of rule-based smart environments. While counterfactual explanations, which describe what could have been done differently to achieve a desired outcome, are a powerful tool in eXplainable AI (XAI), no established methods exist for generating them in these rule-based domains. In this paper, we present the first formalization and implementation of counterfactual explanations tailored to this domain. It is implemented as a plugin that extends an existing explanation engine for smart environments. We conducted a user study (N=17) to evaluate our generated counterfactuals against traditional causal explanations. The results show that user preference is highly contextual: causal explanations are favored for their linguistic simplicity and in time-pressured situations, while counterfactuals are preferred for their actionable content, particularly when a user wants to resolve a problem. Our work contributes a practical framework for a new type of explanation in smart environments and provides empirical evidence to guide the choice of when each explanation type is most effective.**

***Index Terms*—Explanation, Explainable Systems, Counterfactual Explanation, Smart Environment**

I. Introduction

Smart environments, such as smart homes, offices, and buildings, integrate sensor-enabled devices to support users in decision-making, monitoring, and managing abnormal situations [1], [2]. The rapid adoption of these environments is fueled by advances in the Internet of Things (IoT) and Artificial Intelligence (AI), decreasing device costs, and improved system integration [3]–[5].

Rule-based systems are a prevalent approach for implementing automation in smart environments, by executing predefined rules when certain conditions are met [6], [7]. Despite their prevalence, this automation can be difficult for users to interpret. This problem is exacerbated in multi-user environments where rules are created and managed jointly, as users frequently struggle to understand the internal logic driving a particular action or system response.

Providing explanations has been shown to significantly enhance understanding, user perception, and task performance by offering insights into system behavior and highlighting causal factors [8]. Explainability is also strongly linked to trust and transparency: mismatches between user expectations and system behavior erode trust, whereas explanations can mitigate this effect [9]–[12]. Consequently, explainability is now increasingly integrated into a wide range of intelligent systems and application domains [13]–[15].

Among various explanation types, counterfactual explanations are particularly promising. They explain an outcome by describing what would need to have been different for an alternative outcome to have occurred (e.g., “A would have happened if...”) [16]. This form of reasoning is deeply intuitive, mirroring how humans learn and infer causality [17], [18]. Counterfactual explanations are uniquely suited for smart environments for several reasons. First, they excel in controllable and repeatable situations where users want to learn corrective actions [19]. Second, they directly answer the “how to” and “what if” questions that are most useful in proactive systems, rather than just post-hoc “why” questions [9], [20]. Finally, by showing users how to achieve a desired result, they align with the principles of the GDPR’s “Right to Explanation” without needing to expose the entire internal logic of the system [21].

Despite this strong potential, there is currently no formal definition of counterfactual explanations for rule-based smart environments, nor are there established methods for their generation. This paper addresses this research gap with the following contributions:

1) We propose a formal, literature-grounded definition of counterfactual explanations tailored to the context of rule-based smart environments.
2) We present a novel framework for generating these explanations and provide an implementation to demonstrate its feasibility.
3) We conduct a user-centric evaluation of our approach, addressing a significant need for more user studies in the field of counterfactual research [17].

II. Background and Related Work

Automation is reshaping software engineering, replacing manual tasks with intelligent processes across the development

*Corresponding author: mersedeh.sadeghi@uni-koeln.de

2151-0849/25/$31.00 ©2025 IEEE
DOI 10.1109/ASEW67777.2025.00044

lifecycle and application domains [22]–[24]. Smart homes exemplify this shift, highlighting the increasing role of automation in everyday life. Smart environments are sensor-driven systems that autonomously perceive, reason, and act to enhance user comfort [2], [25]. A common implementation strategy involves *rule-based systems* [6], which consist of a knowledge base (rules and device states) and an inference engine that evaluates rule preconditions and fires rules when satisfied [26].

Each rule comprises logical *preconditions* and *actions*. When multiple rules with conflicting actions are simultaneously eligible, a conflict resolution strategy is required. While approaches like specificity or recency exist [27], we adopt *priority-based scheduling* [28], where each rule is assigned a unique priority, and the highest-priority rule is executed.

To support transparency, smart environments can incorporate an *explanation layer* [26]. A system is considered explainable if it provides information that enables users to understand specific outcomes [29]. While causal explanations describe how system logic led to a decision, *counterfactual explanations* describe how an alternative condition could have produced a different outcome [21].

Theoretical foundations for counterfactual reasoning include the *possible worlds* view [30], *structural causal models* [31], and *interventionist accounts* [20], all of which inform definitions used in XAI. Most definitions emphasize *minimality*—changing as little as possible to achieve a different result [17]. In our setting, we define counterfactual explanations as the minimal change to explanation constructs required to achieve a desired alternative outcome.

a) Explainability in Smart Environments.: Several frameworks enable explainability in IoT contexts. *MAB-EX* supports runtime explanation generation for cyber-physical systems [32], while a modular architecture by [33] introduces Local Explanatory Components (LECs) for per-device transparency. Agent-based systems have also been deployed in lab environments [34]. Explainable human activity recognition has been applied in caregiver monitoring systems [35].

SmartEx [7] provides context-aware explanations in rule-based environments and was extended by [36] to support contrastive reasoning.

b) Counterfactual Explanations in XAI.: In XAI, counterfactuals are used to show how minimal input changes lead to different predictions. Several methods ensure feasibility or diversity: *FACE* emphasizes realistic paths, *DiCE* focuses on generating diverse outputs, and *FOCUS* handles tree-based models [37], [38].Other work uses GANs, ASP, or interpretable decision trees [39], [40]. Counterfactuals have also been applied to recommender systems and reinforcement-learned causal models [29].

Despite this broad literature, to our knowledge, no prior work has addressed counterfactual explanations for rule-based smart environments. Our work closes this gap by introducing a formal definition and a practical generation framework tailored to this class of systems.

III. Approach

We define counterfactual explanations for rule-based smart environments by extending established notions of minimal change [17], [40]. Table I lists the core concepts and terminology that underpin our proposed framework.

Definition 1. *A counterfactual explanation in a smart environment is an explanation containing the minimal change to explanation constructs, such that a specific* Foil *would have occurred instead of the* Fact.

Explanation Constructs refer to specifications, facts, propositions, and events relating to both system internals and the external environment [7]. Following contrastive explanation theory [41], [42], we define the **Fact** as the actual outcome prompting an explanation request and the **Foil** as the user-expected alternative outcome.

In the context of smart environments, a user requiring an explanation is typically not interested in any possible alternative to an event, but rather in achieving a specific, desired outcome. This user-centric requirement informs our choice of a counterfactual framework. While many theories [17], [40] define the *Foil* broadly as any event differing from the *Fact*, we align our methodology with that of Wachter et al. [21], which defines the *Foil* as a single, specific incident. This approach ensures the generated explanation is directed toward achieving the user's explicit goal. Within this scope, our framework is concerned solely with constructing an explanation why the observed outcome occurred instead of the specific *Foil* provided by the framework proposed by Herbold et al. [36]; we do not evaluate the validity of the *Foil* itself. The process, therefore, relies on the assumption that this *Foil* accurately represents the user's intended outcome.

Lastly, Counterfactual explanations can be **Additive** or **Subtractive** [43]. In rule-based system, additive explanations aim to fire Appropriate Rule(s) (Table I) to reach the *Foil*, a process associated with creative problem-solving. In contrast, subtractive explanations work by preventing a Disturbing Rule (Table I) from activating, which supports analytical reasoning. [44].

A. Cases for Explanation Needs

We identify three distinct cases of explanation needs for counterfactual explanations in smart environments.

Case E1: An Undesired Event Occurred. The first type of Confusing Situation (Table I) occurs when, at an initial time t_0, the system behaves as expected. At a subsequent time t_1, however, a Disturbing Rule is triggered, causing the Device of Interest (Table I) to transition to an unexpected state. In this case, the *Foil* corresponds to maintaining the state that existed at t_0. Hence, Expected State = Previous State, while the Current State is distinct (Table I).

Case E2: An Expected Event Did Not Occur. This case describes confusion due to system inaction, where the state remains unchanged from t_0 to t_1 (Previous State = Current State). The *Fact* is this persistence, while the *Foil* is

TABLE I: Core definitions used in this work.

Term	Definition
Confusing Situation	A Confusing Situation arises when there is a discrepancy between the observed reality (the *Fact*) and the user's expectation (the *Foil*).
Device of Interest	The Device of Interest is the specific device responsible for the Confusing Situation, i.e., the component of the smart environment whose state contradicts user expectations.
Device of Interest States:	
Previous State	The state of the Device of Interest at an initial time t_0.
Current State	The state of the Device of Interest at a subsequent time t_1, when the need for an explanation arises.
Expected State	The state the user anticipated for the Device of Interest, derived from the foil; used interchangeably with *Foil* in this paper.
Disturbing Rules	The set of Disturbing Rule comprises all rules whose preconditions are satisfied (true) in the current system state. The presence of these enabled rules prevents the system from achieving or maintaining the *Foil*. A conflict–resolution mechanism (e.g., priority) selects one such rule to execute; its execution directly causes the state transition leading to the Confusing Situation.
Appropriate Rule	An Appropriate Rule is a rule whose actions can bring the Device of Interest to the desired state. An Appropriate Rule may itself be *active* or *inactive*.
Rule State (Active / Inactive)	A rule is *active* if all of its preconditions hold in the current system state; otherwise, it is *inactive*.
Rule Priority	A property of each rule used for conflict resolution, ensuring that when multiple rules are active simultaneously, only the one with the highest priority is fired.
Rule Activation	The actions required to satisfy all preconditions of a rule, thereby making it active.
Rule Inactivation	Preventing a rule from being fired, which is achieved by making at least one true precondition false.
Overriding a Rule	Preventing a rule from being fired, by activating another rule with a higher priority.

the transition to the very state of the Device of Interest, that the user anticipated, i.e., Expected State.

Case E3: A Different Event Occurred. This case represents a Confusing Situation where an event occurred (A Disturbing Rule fired) and brought the Device of Interest into a particular state. So, the system has transitioned from a Previous State at time t_0 to a new Current State (the *Fact*) at time t_1, but the user expected a transition to a different state altogether (the *Foil*). Consequently, all three states, Previous State, Current State, and Expected State, are distinct from one another.

B. Foil Achievement Strategy Selection.

Counterfactual explanations focus on "contrary-to-fact" reasoning: they describe how a desired outcome (*Foil*) could have been achieved by analyzing what would need to differ from the current situation [16]. In our framework, this involves first understanding why the Expected State did not occur, identifying what must change to achieve it, and then describing such changes in the form of an explanation.

Table II summarizes the three general strategies for foil achievement in a rule-based system, which depend on the presence or absence of Appropriate Rule(s) and Disturbing Rule(s). In short, the *Foil* may be realized by (i) *Activating* (and ultimately firing, see Table I) some Appropriate Rule(s), (ii) *Inactivating* (Table I) Disturbing Rule(s) that preempt Appropriate Rule, or (iii) a combination of both.

Each Explanation Case (Case E1–Case E3) can map to one or more of these strategies, depending on whether the system lacked active rules, contained preempted rules, or was constrained by reinforcing Disturbing Rule.[1]

a) Additive Counterfactual Explanation: it is a description of how to achieve the Expected State through the Case F1. In this case, all Appropriate Rule(s) with a priority higher than or equal to the highest-priority Disturbing Rule (if any) are considered. For each such Appropriate Rule, the *minimal set of changes* required to activate it is computed (See Section III-C). Since a rule fires only when *all* of its preconditions are satisfied, each false precondition must be resolved. This can be achieved either directly, by modifying the state of relevant devices to satisfy the required preconditions, or indirectly, by firing other rule(s) whose actions enable those preconditions to be met. In the indirect case, the minimal changes needed to fire the supporting rule are determined recursively in the same manner. For each precondition, the framework selects the smaller of the direct or indirect cost. This process is repeated across all false preconditions of the candidate Appropriate Rule, yielding the minimal change set required to activate it. Once this minimal change set is established for each candidate Appropriate Rule, the framework selects the Appropriate Rule with the overall smallest change set. The resulting *Additive Counterfactual Explanation* consists of a description of these minimal changes, which represent the necessary actions to achieve the *Foil*.

[1] For instance, the Case E2 has three sub-cases, each mapping to a different one of the three strategies based on the underlying cause of the system's inaction. However, due to space constraints, we omit a detailed analysis of these mappings in this manuscript. The complete specification is available in our extended work at http://kups.ub.uni-koeln.de/id/eprint/78813.

TABLE II: Foil-achievement cases: reasons for failure, required differences, and resolution strategies

Case	Properties of Factual Situation	What Must Have Been Different to Achieve *Foil*	What Must Be Done to Achieve the *Foil*
F1	No Appropriate Rule was active, and no Disturbing Rule was present.	At least one Appropriate Rule should have been active.	Identify an Appropriate Rule and *Activate* it.
F2	An Appropriate Rule was active, but it was preempted by a Disturbing Rule with a higher priority.	No Disturbing Rule with a higher priority than the Appropriate Rule should have been active.	*Inactivate* all Disturbing Rule that have a higher priority than the Appropriate Rule.
F3	No Appropriate Rule was active, while at least one Disturbing Rule was.	No Disturbing Rule, but, at least one Appropriate Rule should have been active.	All Disturbing Rule must be inactivated and one Appropriate Rule must be identified and activated

b) Subtractive Counterfactual Explanations.: It describes how to achieve the *Foil* by *Inactivating* one or more Disturbing Rule(s). In this case, the minimal change required to falsify at least one precondition of the Disturbing Rule is computed. For each precondition, the framework determines the modification that would make it false and selects the option requiring the smallest change. As in the additive case, such a modification can be implemented either directly or indirectly by firing another rule whose action changes the relevant state. The latter is again evaluated recursively in the same manner as for additive explanations.

An additional complication arises when a precondition to be falsified (e.g., "device d is in state s_1") is continuously reinforced by another rule r_1 with already satisfied preconditions, whose action enforces d$\rightarrow s_1$. In such situations, falsifying the precondition requires not only altering the device state but also inactivating r_1 itself, since otherwise the system would immediately restore d to s_1. The resulting *Subtractive Counterfactual Explanation* therefore provides a description of the minimal set of changes required to *inactivate* the blocking Disturbing Rule (s) and prevent their effects from recurring, thereby enabling the system to achieve the *Foil*.

C. Minimal Change Computation

In line with research on human-centric explanations, our methodology follows the principle of *minimal change*, as humans tend to prefer simple explanations that cite only main causes while ignoring unnecessary details [8], [45]. In the context of counterfactuals, this means providing the smallest set of actions and information required for the user to achieve their *Foil*. Accordingly, we identify five desirable properties that a counterfactual explanation should fulfil (see Table III), which our framework evaluates in a two-phase process.

a) Candidate Filtering with Controllability: Controllability, which measures the user's ability to enact a suggested change, is arguably the most critical property. From a practical standpoint, providing an explanation that relies on uncontrollable changes (e.g., weather conditions) is ineffective. Therefore, our framework uses *Controllability* as a primary filter. Changes that are *immutable* are generally discarded, while *mutable but non-actionable* changes are evaluated by recursively analyzing the controllability of their underlying rule preconditions.

b) Scoring and Ranking with MCDM (Multi-Criteria Decision Making) Method: Candidates that pass the Controllability filter are then scored and ranked based on four quantitative properties: *Sparsity*, *Temporality*, *Proximity*, and *Abnormality*. All four of these properties are conceptually non-beneficial measures, meaning that lower values indicate more desirable minimal changes. Abnormality, however, requires special handling: for subtractive changes, high abnormality is beneficial (removing unusual events), whereas for additive changes the measure is inverted to reflect the "normality" of the desired state. This ensures that abnormality is consistently treated as a beneficial property to be maximized during minimal-change computation.

To aggregate these four criteria, often conflicting, scores and select the optimal candidate, we employ the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution). TOPSIS is a well-established MCDM method chosen for its robustness and widespread use [46], [46], [47], and for consistency with the framework we adapt for *Foil* determination [36]. The candidate ranked highest by TOPSIS constitutes the final minimal change presented in the explanation.

D. Generation of the Counterfactual Explanation

The final step in our framework is to translate the optimal set of minimal changes, identified by the ranking process, into a human-readable explanation. This set contains specific instructions for additive changes (e.g., "device d should have had state s") and general instructions for subtractive changes (e.g., "device d should not have had state s"). To present this information to the user, we employ a natural language template. The template is composed of four key elements: (1) the Device of Interest, (2) its Expected State (the *Foil*), (3) the required additive changes, and (4) the required subtractive changes. To emphasize that the explanation refers to minimal changes that should have occurred, the pattern uses counterfactual conditional tense. The resulting template is shown below:

> *The Device of Interest* would be in the *Expected State* if *additive changes* had happened and *subtractive changes* had not happened.

For example, suppose Alice enters the living room and notices that the lamp is turned on, which surprises her. She expected the lamp to remain off, as she intended to keep the room dark (Expected State = "lamp off"), but instead observes the fact (Current State) that the lamp is on.

TABLE III: Desirable properties of minimal changes in counterfactual explanations

Property	**Description** | **Sources**
Controllability:	The user's ability to implement the suggested changes. The changes are classified as **actionable** (direct user control), **mutable but non-actionable** (indirectly controllable via rules), or **immutable** (no control). | [18], [48]–[50]
Sparsity:	The number of individual changes required to achieve the *Foil*. As users prefer short and simple explanations, lower sparsity is considered better. Our framework uses this property in two ways: first, it is a primary cost factor to be minimized during the computation of the minimal change. Second, it serves as a hard constraint, automatically excluding any potential explanation that requires more than three changes. | [38], [51], [52]
Temporality:	People tend to undo more recent events rather than distant ones. The temporality of a change is defined by the time elapsed since the relevant state last occurred; the shorter the interval, the more favorable the change. If no timestamp exists, a maximum value is assigned to ensure exclusion. Thus, recent changes are prioritized in minimal-change computation, while older ones are penalized. | [53]
Proximity	It counts the number of resulting system changes if all candidate changes were applied. The score is determined by simulating the downstream effects of a change, including any new rules that would subsequently fire or be preempted. The calculation differs based on the explanation type: for additive changes, effects are simulated forward from the new state, while for subtractive changes, the state prior to the prevented event is considered. Lower proximity values indicate more desirable minimal changes. | [38]
Abnormality	Based on the cognitive principle that users focus on altering unusual or exceptional events. This property measures the abnormality of a state based on its historical frequency. | [18], [54]

The system analyzes the situation and finds that two *Active* Disturbing Rule. First, **`[DR-1, Priority 4]:`** `If it is after 5 p.m., turn on the lamp.` Second, **`[DR-2, Priority 2]:`** *`If the sun has set, turn on the lamp.`* Note that, while the only Disturbing Rule with higher priority initially turned on the lamp, both remain active at the moment of explanation.

To achieve the *Foil* (Expected State = "lamp off"), the system finds two Appropriate Rule rules. First, **`[AR-1, Priority 1]:`** `If it is sunny, turn off the lamp.` Second, **`[AR-2, priority 3]:`** `If the room is empty, turn off the lamp.` The framework computes the minimal changes required for any (theoretically) possible *Foil* achievement:

- To *Inactivate DR-2*, that has a higher priority, the condition {sun_set = false} must hold. To *Override DR-1*, either *AR-1* or *AR-2* must fire.
- To override *DR-2* directly, *AR-1* must be fired, which requires the {weather = sunny}.
- Finally, the system can *Inactivate* both DR-1 and DR-2 by setting {time = before 5 p.m., sun_up = true}.

All of these candidates go through the minimality computation. Among all, "making the room empty" is deemed minimal since having a "sunny weather" or "set time before 5 p.m." are *immutable*. The framework selects the minimum change set and generates the following explanation: *"The lamp would have been off if the room had been empty."*

E. Implementation

We implemented our framework as a plugin[2] for *SmartEx* [7], a RESTful Java service with MongoDB, integrated into smart environments via Home Assistant[3]. *SmartEx* provides causal [7] and contrastive explanations [36], and our plugin extends it with counterfactual capabilities. Figure 1 shows the references architecture of our framework.

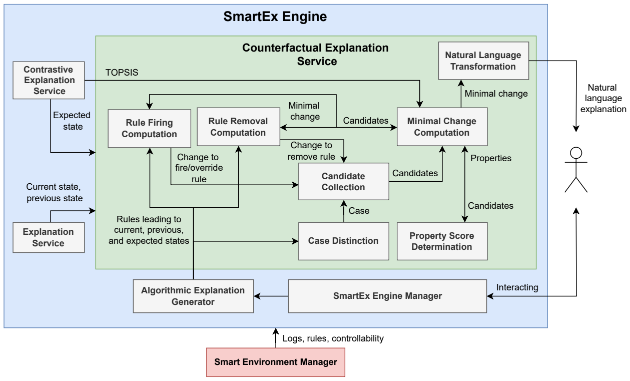

SmartEx Engine
Counterfactual Explanation Service
Natural Language Transformation
Contrastive Explanation Service
TOPSIS
Expected state
Rule Firing Computation
Rule Removal Computation
Minimal change
Candidates
Minimal Change Computation
Minimal change
Natural language explanation
Change to fire/override rule
Change to remove rule
Candidate Collection
Candidates
Properties
Current state, previous state
Case
Candidates
Explanation Service
Rules leading to current, previous, and expected states
Case Distinction
Property Score Determination
Algorithmic Explanation Generator
SmartEx Engine Manager
Interacting
Logs, rules, controllability
Smart Environment Manager

Fig. 1: Reference architecture of *SmartEx* and the *Counterfactual Explanation Service*

When a counterfactual explanation is requested, *SmartEx* identifies the Device of Interest and fetches all relevant states (Current State, Previous State) as well as all *Active Rules* related to the Device of Interest. It then interacts with the existing *Contrastive Explanation* plugin for *Foil* and Expected State determination.

Given that, the *Case Distinction Component* then identifies the explanation case (Case E1–Case E3). The *Candidate Collection Component* gathers all Disturbing Rule(s) and Appropriate Rule(s), computes the best *Foil* achievement strategy, and creates combinations of how candidate rules can be *Activated*, *Inactivated*, or *Overridden*. The *Minimal Change Computation Component* removes duplicates and excludes all candidates that are not *actionable* if fully *actionable* ones exist. Finally, candidates are scored for sparsity, temporality, proximity, and abnormality. Using TOPSIS, the best candidate is selected and transformed into natural language. The generated explanation is finally delivered to the user through *SmartEx*'s smartphone or web application.

[2] https://github.com/ExmartLab/SmartEx-Engine/tree/counterfactual
[3] https://www.home-assistant.io/

IV. Evaluation

To assess how our generated counterfactual explanations are received in practice, we conducted a quantitative, human-centered evaluation. The study was designed to compare counterfactual explanations against traditional causal ones, guided by two research questions (RQ):

- **RQ1:** Do users prefer counterfactual or causal explanations in smart environments?
- **RQ2:** In which contexts do users prefer counterfactual or causal explanations in smart environments?

A. Study Design

We conducted a study using a within-subject experimental design with 17 participants, recruited via personal contacts. Each participant took part in a 15-minute, in-person interview session. After a brief welcome and introduction, participants were informed about data privacy and asked two preliminary questions to assess their general preferences: (1) short vs. detailed explanations, and (2) explanations that provide reasons vs. those that offer solutions. These baseline responses were later used to contextualize participants' preferences (RQ2). Participants then experienced two narrative-driven scenarios comprising a total of six *confusing scenes* caused by automation. Each scene was presented through a slide-based illustration of a 3D-style top-down room view, with clearly depicted devices (e.g., speakers, blinds, air conditioning), brief textual descriptions, and sound or visual cues (e.g., musical notes or brightness level changes) to support the user's mental model and enhance immersion.

Before each scene, participants were reminded that all upcoming explanations were factually correct to avoid bias in evaluation. After encountering the confusing situation, they were asked whether they *wanted* an explanation (choosing between "yes", "I don't care", or "no"). This question was included solely for analytical purposes but it did not affect the study procedure. Regardless of their answer, all participants were then provided with three paper snippets: a *causal explanation*, a *counterfactual explanation*, and a *no explanation* option. They were asked to rank these options based on their subjective preference.

Scenes varied across several dimensions[4]: environmental setting (home vs. office), urgency (scenes with and without time pressure), and the degree to which *causal and counterfactual explanations diverged*. In some cases, such as Scene 2, counterfactual explanations were concise and actionable, proposing specific changes to resolve the situation (Table IV). In contrast, in some scenes such as Scene 5, where no actionable change was possible (i.e., all relevant conditions were *immutable*), the counterfactual and causal explanations were functionally equivalent, differing only in linguistic framing.

After all six scenes were completed, participants were formally introduced to the distinction between **causal** and **counterfactual** explanations, and were provided with a reference list of all explanations encountered during the study. Finally, participants completed a 4-item Likert-scale questionnaire, rating each explanation type (causal and counterfactual) on (1) linguistic clarity and (2) content usefulness.

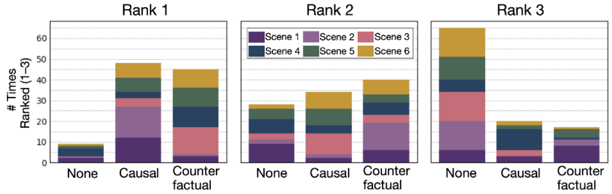


Fig. 2: Distribution of explanation type rankings across six scenes and 17 participants. It shows how often each explanation type was ranked 1st, 2nd, or 3rd across all participant–scene combinations (total: 102 per rank).

B. Results and Discussion

To investigate users' preferences for different explanation types in smart environments, we analyzed their rankings and ratings across six interactive scenes. Regarding **RQ1**, we found no strong overall preference for either explanation type. As shown in Figure 2, *causal explanations were ranked first slightly more often* than counterfactual ones, but they were also ranked third more frequently, indicating more polarized opinions. Causal explanations were especially favored in the two initial scenarios, which involved time-sensitive or straightforward automation events. In contrast, counterfactual explanations were preferred in the remaining scenarios, particularly those without time pressure or where a suggested action could alter the system's behavior. Across all scenes, the "no explanation" option was generally least preferred. This lack of a clear overall winner suggests that user preference is not absolute and is instead driven by specific contexts, which we explore in our analysis of RQ2.

To answer **RQ2**, we analyzed how preferences varied across different contexts:

Explanation-Specific Factors: The post-study questionnaire revealed that the evaluation criterion is a primary contextual factor. Participants expressed a strong preference for causal explanations *linguistically* (Mean 4.24 vs. 2.94). However, when judging the *content*, they rated counterfactual explanations slightly more favorably (Mean 3.65 vs. 3.53). This crucial distinction shows that while users find the language of causal explanations easier to parse, they often find the information within counterfactuals more valuable.

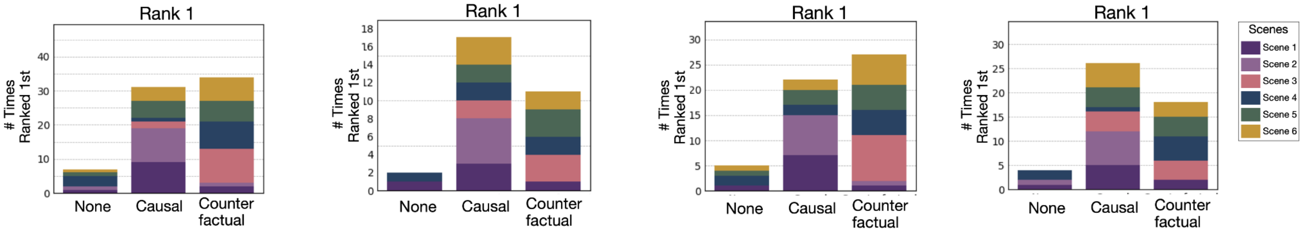
User's Goal and Style: The user's preexisting preferences strongly correlated with their choices (see Figure 3). Participants who prefer *shorter, solution-oriented explanations* showed a clear preference for *counterfactuals*. Conversely, those who prefer *longer, reason-oriented explanations* strongly favored *causal* ones.

Situational Context: The situation in each scene also had a significant impact. We found that *causal* explanations were more preferred when users were under *time pressure*, suggesting they require less cognitive effort to comprehend. In

[4] Due to space limitations, we do not provide descriptions of the scenes here. Please refer to the extended version (http://kups.ub.uni-koeln.de/id/eprint/78813) for complete details.

TABLE IV: Explanations provided to participants

Scene	Exp. Type	Explanation
1	Causal Counterfactual	The speaker is on because no meeting is going on in a meeting room, and the social room is not empty. The speaker would be off if there was a meeting going on in a meeting room.
2	Causal Counterfactual	The meeting room door is locked because it is before 8:30 a.m. The meeting room door would be open if it was not before 8:30 a.m.
3	Causal Counterfactual	The brightness is at 70% because there is only a single person in the room. The brightness would be at 100% if a device was connected to the beamer.
4	Causal Counterfactual	The speaker remains off because no rule was executed. The speaker would be on if there was no meeting going on.
5	Causal Counterfactual	The air conditioning is on because it is sunny and all windows are closed. The air conditioning would be off if the door was open longer than 10 minutes and not all windows were closed.
6	Causal Counterfactual	The blinds are rolled down halfway because the blind's controller down button was pressed twice, and the plant lights are off. The blinds would be rolled down completely if the plant lights were not off.



(a) Shorter Explanation (b) Longer Explanation (c) Solution Explanation (d) Reasoning Explanation

Fig. 3: Number of times each explanation type was ranked 1st across all scenes, grouped by participants' stated explanation preference.

contrast, *counterfactual* explanations were strongly preferred when users expressed a high *need for an explanation* and a *desire to change* the situation.

Explanation-Specific Properties: The actionability of the explanation was highly influential. The preference for counterfactual explanations was significantly stronger when they were *actionable*, validating our framework's focus on this property. Furthermore, among the counterfactual explanations, those with *subtractive structure*, emphasizing the inactivation of a triggering cause, were slightly more preferred than additive ones that introduced hypothetical alternatives. This may be because subtractive explanations align more closely with users' intuitive causal reasoning and focus directly on what went wrong.

Overall, explanation preferences were context-dependent. Causal explanations were preferred in time-pressured or low-engagement contexts. Counterfactual explanations were valued when users desired control, actionable suggestions, or deeper understanding. These findings support the case for offering both types dynamically, tailored to situational and user-specific factors in smart environments.

V. Conclusion

In this paper, we addressed the lack of counterfactual explanations in rule-based smart environments by proposing a formal definition and a novel generation framework. Our approach operationalizes the principle of "minimal change" by scoring candidate explanations against five desirable properties (e.g., controllability, sparsity). We implemented and evaluated this framework in a human-centered study, which confirmed that user preference is highly contextual, revealing a trade-off between the linguistic simplicity of traditional causal explanations and the actionable, solution-oriented content of our counterfactuals. The key implication is that intelligent systems should adapt the explanation type to the user's context and goals rather than relying on a single method. While this study has limitations, our framework provides a foundation for future work, particularly in using large language models to improve the linguistic quality of counterfactuals and in conducting larger, in-situ validations.

References

[1] D. M. El-Din, A. E. Hassanein, and E. E. Hassanien, "Smart environments concepts, applications, and challenges," *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*, pp. 493–519, 2021.

[2] E. Ahmed, I. Yaqoob *et al.*, "Internet-of-things-based smart environments: State of the art, taxonomy, and open research challenges," *IEEE Wireless Communications*, vol. 23, no. 5, 2016.

[3] W. Li, T. Yigitcanlar, I. Erol, and A. Liu, "Motivations, barriers and risks of smart home adoption: From systematic literature review to conceptual framework," *Energy Research & Social Science*, vol. 80, 2021.

[4] L. Baresi, M. Sadeghi, and M. Valla, "Tdex: A description model for heterogeneous smart devices and gui generation," in *2018 IEEE International Conference on Internet of Things (IThings)*. IEEE, 2018.

[5] L. Baresi and M. Sadeghi, "Fine-grained context-aware access control for smart devices," in *2018 8th International Conference on Computer Science and Information Technology (CSIT)*. IEEE, 2018, pp. 55–61.

[6] C. Nandi and M. D. Ernst, "Automatic trigger generation for rule-based smart homes," in *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, 2016, pp. 97–102.

[7] M. Sadeghi, L. Herbold, M. Unterbusch, and A. Vogelsang, “SmartEx: A framework for generating user-centric explanations in smart environments,” in *2024 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2024, pp. 106–113.

[8] L. Chazette and K. Schneider, “Explainability as a non-functional requirement: Challenges and recommendations,” *Requirements Engineering*, vol. 25, no. 4, pp. 493–514, 2020.

[9] B. Y. Lim, A. K. Dey, and D. Avrahami, “Why and why not explanations improve the intelligibility of context-aware intelligent systems,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2009, pp. 2119–2128.

[10] M. Sadeghi, V. Klös, and A. Vogelsang, “Cases for explainable software systems: Characteristics and examples,” in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2021.

[11] M. Winikoff, “Towards trusting autonomous systems,” in *Engineering Multi-Agent Systems: 5th International Workshop, EMAS*. Springer, 2018, pp. 3–20.

[12] M. Sadeghi, D. Pöttgen, P. Ebel, and A. Vogelsang, “Explaining the unexplainable: the impact of misleading explanations on trust in unreliable predictions for hardly assessable tasks,” in *Proc. of the 32nd ACM Conf. on User Modeling, Adaptation and Personalisation*, 2024.

[13] T. Sakai and T. Nagai, “Explainable autonomous robots: a survey and perspective,” *Advanced Robotics*, vol. 36, no. 5-6, pp. 219–238, 2022.

[14] M. Unterbusch, M. Sadeghi, J. Fischbach, M. Obaidi, and A. Vogelsang, “Explanation needs in app reviews: Taxonomy and automated detection,” in *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*. IEEE, 2023, pp. 102–111.

[15] D. J. Yeong, K. Panduru, and J. Walsh, “Exploring the unseen: A survey of multi-sensor fusion and the role of explainable ai (xai) in autonomous vehicles,” *Sensors*, vol. 25, no. 3, p. 856, 2025.

[16] I. Stepin, J. M. Alonso, A. Catala, and M. Pereira-Fariña, “A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence,” *IEEE*, vol. 9, 2021.

[17] R. Guidotti, “Counterfactual explanations and how to find them: Literature review and benchmarking,” *DMKD*, pp. 1–55, 2022.

[18] R. M. Byrne, “Counterfactuals in explainable artificial intelligence (XAI): Evidence from human reasoning.” in *Proc. of the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2019.

[19] N. J. Roese, “Counterfactual thinking.” *Psychological Bulletin*, vol. 121, no. 1, p. 133, 1997.

[20] J. Woodward, *Making things happen: A theory of causal explanation*. Oxford University Press, 2003.

[21] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual explanations without opening the black box: Automated decisions and the GDPR,” *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[22] M. Ryalat, H. ElMoaqet, and M. AlFaouri, “Design of a smart factory based on cyber-physical systems and internet of things towards industry 4.0,” *Applied Sciences*, vol. 13, no. 4, p. 2156, 2023.

[23] S. Kalwar, M. Sadeghi, A. J. Sabet, A. Nemirovskiy, M. Rossi *et al.*, “Smart: Towards automated mapping between data specifications,” in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, vol. 2021. Knowledge Systems Institute Graduate School, 2021, pp. 429–436.

[24] M. Hosseini, S. Kalwar, M. Rossi, M. Sadeghi *et al.*, “Automated mapping for semantic-based conversion of transportation data formats,” in *CEUR Workshop Proceedings*, vol. 2447, 2019, pp. 1–6.

[25] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, “A review of smart homes-Past, present, and future,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, 2012.

[26] N. Masri, Y. A. Sultan, A. N. Akkila, A. Almasri, A. Ahmed, A. Y. Mahmoud, I. Zaqout, and S. S. Abu-Naser, “Survey of rule-based systems,” *International Journal of Academic Information Systems Research (IJAISR)*, vol. 3, no. 7, pp. 1–23, 2019.

[27] R. Ali and M. o. Afzal, “Knowledge-based reasoning and recommendation framework for intelligent decision making,” *Expert Systems*, vol. 35, no. 2, 2018.

[28] T. Shah, S. Venkatesan, T. Ngo, and K. Neelamegam, “Conflict detection in rule based IoT systems,” in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2019, pp. 0276–0284.

[29] P. Madumal, T. Miller, L. Sonenberg, and F. Vetere, “Explainable reinforcement learning through a causal lens,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 03, 2020.

[30] D. Lewis, “Counterfactuals and comparative possibility,” in *IFS: Conditionals, Belief, Decision, Chance and Time*. Springer, 1973, pp. 57–85.

[31] J. Pearl, “Models, reasoning and inference,” *Cambridge University Press*, vol. 19, no. 2, p. 3, 2000.

[32] M. Blumreiter, J. Greenyer, F. J. C. Garcia, V. Klös, M. Schwammberger, C. Sommer, A. Vogelsang, and A. Wortmann, “Towards self-explainable cyber-physical systems,” in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 2019, pp. 543–548.

[33] E. Houzé, A. Diaconescu, J.-L. Dessalles, and D. Menga, “A generic and modular reference architecture for self-explainable smart homes,” in *2022 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. IEEE, 2022, pp. 101–110.

[34] A. Dobrovolskis, E. Kazanavičius, and L. Kižauskienė, “Building XAI-based agents for IoT systems,” *Applied Sciences*, vol. 13, no. 6, 2023.

[35] D. Das, Y. Nishimura *et al.*, “Explainable activity recognition for smart home systems,” *ACM Transactions on Interactive Intelligent Systems*, vol. 13, no. 2, 2023.

[36] L. Herbold, M. Sadeghi, and A. Vogelsang, “Generating context-aware contrastive explanations in rule-based systems,” in *Proceedings of the 2024 Workshop on Explainability Engineering*, 2024, pp. 8–14.

[37] A. Lucic, H. Oosterhuis *et al.*, “FOCUS: Flexible optimizable counterfactual explanations for tree ensembles,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[38] R. K. Mothilal, A. Sharma, and C. Tan, “Explaining machine learning classifiers through diverse counterfactual explanations,” in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 607–617.

[39] J. Van der Waa, M. Robeer, J. van Diggelen, M. Brinkhuis, and M. Neerincx, “Contrastive explanations with local foil trees,” *Workshop on Human Interpretability in Machine Learning (WHI)*, pp. 41–46, 2018.

[40] L. Bertossi, “An ASP-based approach to counterfactual explanations for classification,” in *Rules and Reasoning: 4th International Joint Conference*. Springer, 2020, pp. 70–81.

[41] P. Lipton, “Contrastive explanation,” *Royal Institute of Philosophy Supplements*, vol. 27, pp. 247–266, 1990.

[42] T. Miller, “Contrastive explanation: A structural-model approach,” *The Knowledge Engineering Review*, vol. 36, 2021.

[43] N. J. Roese and K. Epstude, “The functional theory of counterfactual thinking: New evidence, new challenges, new insights,” in *Advances in experimental social psychology*. Academic Press, 2017, vol. 56.

[44] K. D. Markman, M. J. Lindberg, L. J. Kray, and A. D. Galinsky, “Implications of counterfactual structure for creative generation and analytical problem solving,” *Personality and Social Psychology Bulletin*, vol. 33, no. 3, pp. 312–324, 2007.

[45] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[46] H. Taherdoost and M. Madanchian, “Multi-criteria decision making (MCDM) methods and concepts,” *Encyclopedia*, vol. 3, no. 1, 2023.

[47] C.-L. Hwang and K. Yoon, “Methods for multiple attribute decision making,” *Multiple attribute decision making: Methods and applications*, pp. 58–191, 1981.

[48] A.-H. Karimi, B. Schölkopf, and I. Valera, “Algorithmic recourse: From counterfactual explanations to interventions,” in *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 2021.

[49] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, “A survey of methods for explaining black box models,” *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[50] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez *et al.*, “FACE: Feasible and actionable counterfactual explanations,” in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020.

[51] S. Verma, V. Boonsanong *et al.*, “Counterfactual explanations and algorithmic recourses for machine learning: A review,” *ACM Computing Surveys*, vol. 56, no. 12, 2024.

[52] J. Dai, C. Zhang, D. Aliakseyeu *et al.*, “The effect of explanation design on user perception of smart home lighting systems: A mixed-method investigation,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.

[53] D. T. Miller and S. Gunasegaram, “Temporal order and the perceived mutability of events: Implications for blame assignment.” *Journal of personality and social psychology*, vol. 59, no. 6, p. 1111, 1990.

[54] D. Kahneman and A. Tversky, *The simulation heuristic*. Cambridge University Press, 1982, pp. 201–208.