Human-centric Computing and Information Sciences

October 2022 | Volume 12

www.hcisjournal.com

KIPS
Korea Information Processing Society

KIPS CSWRG
Korea Information Processing Society
Computer Software Research Group

HCIS | Human-centric Computing and Information Sciences

RESEARCH (Research Manuscript) **Open Access**

Human-centric Computing and Information Sciences (2022) 12:48
DOI: https://doi.org/10.22967/HCIS.2022.12.048
Received: January 4, 2022; Accepted: May 8, 2022; Published: October 30, 2022

Improving Mental Models in IoT End-User Development

Massimo Zancanaro[1,2], Giuseppe Gallitto[2,3], Dina Yem[1], and Barbara Treccani[1,*]

Abstract

This paper describes two empirical research studies that investigated how to improve naïve users' mental models to support end-user development (EUD) of Internet-of-Things (IoT). Specifically, we intended to evaluate the effectiveness of two different strategies, namely nudging and informing, to support trigger-action (TA) rule programming. To this aim, we analyzed non-expert users' performance and their verbal reports (Studies 1 and 2, respectively) in a task requiring the identification of the outcomes of the execution of specific sets of TA rules in different IoT scenarios. The triggering part of TA rules typically involves instantaneous and/or protracted events, and previous studies have shown that users' poor understanding of the distinction between these two types of events, as well as of the way in which the rules interact with each other, can result in poor TA programming performances. The first (experimental and quantitative) study shows that a nudging strategy (i.e., using two different temporal conjunctions, WHEN and WHILE, to introduce the rules' triggering conditions that refer to the two types of events instead of using the more common and generical IF) improves participants' understanding of the rules' behavior. It also provides some evidence that an informing strategy (i.e., providing participants with an explicit description of how the rules are evaluated and activated) can improve participants' accuracy in identifying the rules that did not realize the desired situation. The second (observational and qualitative) study suggests that the use of WHEN and WHILE in the triggering part of the rule helps participants distinguish the two types of events and understand their semantics. This work extends the current literature in EUD by providing both critical information about users' mental models in IoT and useful suggestions to make appropriate (linguistic and structural) choices when designing the interface that guides users in defining the rules.

Keywords

End-User Development, Internet of Things, Trigger Action Programming, Human-Computer Interaction, Human Factors in Computing Systems

1. Introduction

As the Internet of Things (IoT) is pushing for digitalizing everyday objects [1], it becomes increasingly important to explore new means for users to control sensors and devices [2]. End-user development (EUD) is defined as the possibility for people without programming experience to create or modify their applications [3]. In this respect, it provides an interesting approach to dealing with the IoT [4]. "Smarter"

***Corresponding Author:** Barbara Treccani (barbara.treccani@unitn.it))
[1] Department of Psychology and Cognitive Science, University of Trento, Trento, Italy
[2] Fondazione Bruno Kessler – FBK, Trento, Italy
[3] The Predictive Neuroimaging Lab, Essen University Hospital, Essen, Germany

objects are often less easily accepted by users [5] and the possibility for naïve, non-expert users to actively control them might be a key to acceptance [2]. While user-centered design advocates for users' involvement in the design phases, EUD calls for empowering users beyond these phases and proposes that design, learning, and development are inherent parts of the technology in use [6, 7].

The effectiveness of EUD in the context of IoT-based smart devices has been well demonstrated by the success of initiatives like IFTT [8]. This popular web-based service allows users to create conditional statements triggered by changes in either devices or web apps. This metaphor is readily applicable to IoT [9] since IoT devices are usually either sensors that detect events in the world or actuators that operate changes in the world (or both). The IFFT approach is an example of a programming approach based on contextual rules that have evolved in the so-called trigger-action programming (TAP). A trigger-action (TA) rule takes the specific form of an action that is performed upon the occurrence of an event. Several commercial tools use a similar approach—for example, Amazon's Alexa with the so-called Alexa Routines [10].

Indeed, programming is complicated because it often requires expressing solutions in ways that are not familiar to non-experts [11]. The concept of TA rule provides an intelligible metaphor for the programming of digital technologies because it embeds the idea that specific actions must be taken in specific situations [12].

However, the simplicity of this event-action paradigm is also its limitation. In a study conducted with over 300 MTurk workers [13], the authors collected 1,590 trigger-action programs in the domain of a smart home. Their analysis revealed that 77.9% of program behaviors could be expressed with rules involving single triggers and single actions, but 16.9% required multiple triggers and possibly multiple actions (the remaining 5.2% required a single trigger but multiple actions). To allow effective programming of IoT devices, people need more expressive triggering conditions and more elaborate actions than those provided for by common single trigger- single action TA rules.

Actually, several research prototypes [2, 13–15] and some commercial tools (e.g., SmartThings and SharpTools [16, 17]) permit complex triggering conditions, with multiple triggering events, and multiple actions. Indeed, the TAP paradigm inherits from the so-called ECA (event-condition-action) rules that expert programmers use as a framework for effective control of databases [18, 19] and workflows [20, 21]. In the ECA rules, the condition part can be quite complex (i.e., it may not be limited to the check of the occurrence of an event), and the action part can have the form of an elaborate routine. That allows effective control of the flow of operations while maintaining a fully expressive programming power [21, 22].

However, understanding complicated conditions is problematic for end-users [23, 24]. When the triggering conditions become more complex, the simplicity of the rule-based metaphor drastically diminishes, and users are more prone to errors. The inaccurate composition of events is among the most common errors [25, 26].

Some authors have tried to introduce a simplified version (with a fixed, simple structure) of the condition part of the ECA rules in TA rules. For example, Truong et al. [27] suggest limiting the condition to a syntactical specification of the location (WHERE) in which the event should take place in order for the action to be executed; similarly, the tool EFESTO-5W [28] only supports the specification of temporal (WHEN) and/or spatial (WHERE) aspects concerning the event in the triggering condition. In the present study, we aim to investigate the effectiveness of an approach that provides for constraining the condition part of TA rules, while allowing a richer expressivity than that of the "structural" approaches described above. In doing so, we focus on the difference between two types of events (i.e., instantaneous vs. protracted events) and propose a specific linguistic frame to nudge users to understand and use this distinction.

The distinction between these types of events is grounded in the semantics of natural language and often codified in lexical choices [29, 30]. Events are often specifically conceptualized as properties of moments. Instantaneous events are timeless (for example, "to catch a flu"). In contrast, protracted events have a duration (for example, "the presidential campaign") but they are characterized by undefined or fuzzy time boundaries [31].

In the field of EUD programming for IoT environments, this distinction has been examined by Huang and Cakmak [25], who called instantaneous events simply “events” and protracted events “states.” They proposed that state-based programming might be exploited as an alternative to (or in combination with) event-driven programming. However, they also noted that this distinction might be problematic for the user to understand. Support for clarifying the distinction between events and states at the graphical interface level has been proposed but not further developed by Mattioli and Paterno [32].

We propose exploiting natural language to help users understand the difference between events and states (for the sake of simplicity, we used this terminology rather than instantaneous and protracted events).

Indeed, several languages use different conjunctions to introduce longer events (i.e., the “states” in our terminology, e.g., “while” in English, “während” in German, “mientras” in Spanish, and “mentre” in Italian) and short events (i.e., the “events” proper, e.g., “when” in English, “als” in German, “cuando” in Spanish, and “quando” in Italian). In several cases, these pairs of conjunctions can be used interchangeably. However, in multi-clause sentences, the while clause usually describes the longer event that represents the ground in which the shorter event (in the when clause) is interpreted [33].

Therefore, we propose to express TA rules in the form WHEN <event> WHILE <set of states> THEN <list of actions>. The <event> part of the rule specified a single event. The <set of states> part is a conjunction of logical propositions on the world that can be evaluated as true or false; the set can be empty (in other words, the WHILE part can be omitted). The <list of actions> part is a sequence of actions executed in the specified order, only if the conjunction of logical propositions holds when the specified event occurs.

As Huang and Cakmak [25] suggested, we hypothesize that this structure for TA rules induced a more effective mental model of the system in naïve users.

Indeed, the primary source of confusion in interacting with an artifact is due to users having a wrong or inaccurate mental model of the actual functioning of the system [34]. Mental models are internal representations of (parts of) the world that explain and regulate how people interact with the world [35]. A mental model of a complex artifact is a representation of the mechanism and working of the artifact that is developed by the user to make sense of the artifact itself and to effectively use it [34, 36]. The understanding of users’ mental models is critical for the comprehension of the interaction between users and the artifact.

A user’s mental model does not need to be complete and accurate, but it should represent the core mechanisms of the artefact. Proper design can (and should) implicitly induce effective mental models in the user of a given system [34]. However, an adequate representation of how the system works can also be explicitly communicated. How the system is described to the users can have a strong impact on the users’ mental models of that system, and this, in turn, may result in different user-system interactions. For example, Halasz and Moran [37] proposed two different, albeit both correct, descriptions of the functioning of a reverse-polish calculator to two groups of participants and showed that these descriptions led to different levels of performance.

One aspect that often confuses non-programmers concerns how constructs are expressed in programming languages. For example, Pane and Myers [11] noted that “then” is often interpreted as “afterward” instead of “in these conditions.” Therefore, it is crucial for designing EUD systems to consider how naive users interpret the language used to express the conditions in TA rules. We argue that the form “IF-THEN,” although supposedly simple, does not help naïve users understand the needed complexity of TA rules. In contrast, the “WHILE-WHEN-THEN” form might be more effective in suggesting the differences between events and states by nudging users toward a more effective mental model.

In a seminal work, Brackenbury et al. [26] report several bugs, many of which can be related to the confusion between events and states. Such a confusion alone does not obviously account for all the problems with the more complex forms of TA rules. Another relevant aspect is the proper understanding of the temporality of the rule mechanism. That is, the fact that the rules are cyclically applied. The lack of understanding of the cyclical mechanism determines what is called “repeated triggering” bug: when an action is conditioned on a state whose duration is longer than a single cycle, the rule can be triggered repeatedly and, therefore, the action performed several times (for example, "IF I come within 1 mile of a pizza shop THEN order me a pizza" ends up in ordering many pizzas [26]). We suggest that naïve users might not be fully aware of the cyclical mechanism and may form an inaccurate mental model of the system. In this work, we propose that a more explicit description of this mechanism should be provided to users, rather than simply describing the form of the rules (following the example by Halasz and Moran [37], discussed above). In our view, TAP systems do not (always) need to be walk-up-and-use tools and a (possibly short) learning phase is beneficial and often inevitable. Accordingly, it is critical to understand how to instruct users properly. This work is a first step in such a direction.

In summary, although there is a wide agreement that better mental models of TAP can improve the effectiveness of EUD, there is not much evidence of how this can be done. In order to investigate this issue, we conducted two separate but strictly related studies. They were aimed to analyze naïve users' behaviors in a task that required the identification of the outcomes of the execution of given sets of TA rules in different fictitious scenarios involving a TAP system that rules an automated "smart home." In Study 1 we analyzed participants' performance in this task, whereas in Study 2 we analyzed participants' verbal reports while they were performing the task. We investigated the effectiveness of two different strategies to improve users' mental models of either the TAP system or the specific user-system interactions involved in the different scenarios: (1) a nudging strategy consisting of a language-based manipulation of the TA rules and (2) an informing strategy that consists of clarifying the iterative operational nature of rule-based systems (i.e., the cyclical mechanism). The original contribution of our research lies in providing evidence that naïve users can indeed create more effective TAP mental models when these two strategies are used. In this respect, our results support and extend the recent literature in the field of EUD [23–28, 32].

2. Related Work

IoT is a recent approach to infrastructure information technology that provides a framework to instantiate and leverage other emerging technologies, such as edge computing [38]. IoT environments consist of a large set of resource-constrained devices (from simple sensors to smartphones) with independent identities that operate in orchestrated ways to accomplish large and pervasive tasks. Recently, the metaphor of social networks has been proposed to account for the communication complexity arising from IoT structures [39]. IoT poses several problems both at technological and socio-technical levels [40]. Among the latter type of problems, security, privacy, and trust issues are particularly important. They require not only new architectural and modelling approaches [41], but also new approaches to interact with end-users [5]. EUD can be leveraged as a new framework for a more responsible use of IoT [2]. Since its very beginning, EUD has been proposed as an approach for empowering users in their relationship with technology [6]. More recently, it has been recognized that the distinction between developers and end-users is not straightforward since end-users interested in customizing their tools may range from totally naïve users (even children programming their toys [42]) to technical operators with high programming skills [43].

In the last years, several approaches have been proposed to allow end-users with different programming expertise to program heterogeneous sets of devices. These approaches can be classified according to (1) the extent to which they can be used in different domains; (2) their coverage of either the interactive or functional part of an application; and (3) the extent to which the implementation details are hidden [7].

Those approaches that mainly cover the functional aspects of programming often use a flow-based approach to model the structure of the task (e.g., [44]), while those focusing on the interactive part of the application often rely on an event-driven paradigm [4, 9, 11, 15, 45]. In several domains, the most important aspect for users is controlling the interaction with their devices; therefore, event-driven approaches are the most used in EUD [7]. Flow-diagrams and event-driven approaches might be combined in a single tool (e.g., [46, 47]). Block-based programming [48] has been largely used for presenting either flow-diagrams or event-driven paradigms to end-users in IoT contexts [49] and has demonstrated good usability for non-programmers [4, 50].

In order to tackle more complex problems, the flow-diagrams approach has been recently extended in so-called skills-based programming. A skill is a composition of sensing and manipulation primitives that expert programmers define. Specific tasks can be composed by non-programmers applying the skills to their devices [51].

Other approaches include using natural language instructions [52, 53] and the combination of natural language instructions with block-based event-driven instructions [47]. In order to support users in expressing instructions and representing the sensors and objects to which these instructions apply, techniques of augmented reality [54] and tangible interaction [55] have also been proposed.

In this work, we adopted an event-driven approach based on ECA rules that might be easily adapted to environments wherein these techniques are employed. We used an authoring tool similar to those presented in previous studies [32, 28, 56], but with specific attention to the language used to express the instructions (i.e., the ECA rules). We did not employ block-based programming, and, in this respect, our approach is close to those leveraging on natural language.

3. Study 1: An Experimental Investigation of Participants' Performance in a (Fictitious) TAP Task

Study 1 aimed to evaluate two different hypotheses. In line with the evidence discussed above, we posited that expressing TA rules in a linguistic form that nudges the event/state distinction improved the understanding of the effects of the rules (Hypothesis 1). Specifically, we proposed the following format: WHEN <event> WHILE <set of states> THEN <list of actions>. The <event> part of the rule specified a single event. The <set of states> part was a conjunction of logical propositions on the world that could be evaluated as true or false; the set could be empty (in other words, the WHILE part could be omitted). The <list of actions> part was a sequence of actions executed, in the specified order, only if the conjunction of logical propositions held true and when the specified event occurred. In order to evaluate this hypothesis, we planned to compare participants' performances when they had to deal with rules that have the WHEN/WHILE/DO format with that observed when they faced rules having the (more common) IF/DO format.

We also posit that an explicit description of the cyclical mechanism of evaluation and activation of the rules might prevent some of the bugs in TAP from occurring (Hypothesis 2). In particular, we hypothesized that this description should prevent bugs related to temporality [26]. Accordingly, we decided to compare two descriptions of how a TAP system works: a richer description in which the cyclical nature of the rules' evaluation-and-activation mechanism was made explicit (i.e., a depiction of the system communicating a *Computational* model of this system) and a simpler description of the possible rule structures and of the possible combinations of logical propositions within the rules (i.e., a depiction of the system communicating a *Descriptive* model of this system).

With the aim of assessing these two hypotheses, Study 1 was designed as a controlled experiment in which participants were presented with eight scenarios, each describing an intended goal, together with a set of rules which were supposed to achieve that goal. In some cases, the rules were correct (i.e., they correctly achieved the intended goal). In contrast, in other cases, they were "buggy" (i.e., conditions described in the scenario did not activate the rules, or their activation had outcomes other than those intended). For each scenario, the participants had to assess whether the rules were correct or not and express their confidence in their assessment.

The experiment had two *between-participants* conditions (*Computational* vs. *Descriptive* depictions of the rules' evaluation and activation mechanism) and two *within-participants* conditions (WHILE-WHEN-THEN vs. IF-THEN structures of the rules). The results of a smaller study, which was used as a pilot for the present one, have been published in Gallitto et al. [57]

Following our hypotheses, we expected that mental models induced by the WHEN-WHILE-THEN rule structure and by the *Computational* description were more likely to correctly represent the distinction between events and states and the changes of the rule triggering conditions over time (i.e., the temporality of the TAP system), respectively. That, in turn, should result in more accurate performances. Specifically, following Hypothesis 1, we expected participants to be more accurate when facing rules with the WHEN-WHILE-THEN structure than when they dealt with the IF-THEN structure. Furthermore, according to Hypothesis 2, we expected better performances in the case of participants to whom the *Computational*, rather than *Descriptive*, depiction of the TAP system was given. In particular, the "computational" representation of this system should help participants to detect bugs (e.g., the "repeated triggering" bug) in the buggy scenarios, which critically depended on the understanding of the cyclical nature of the rules' evaluation and activation mechanism.

Conversely, no significant differences between performances observed with the two rule structures or with the two system descriptions were expected if these manipulations were ineffective in eliciting more appropriate mental models of the single rules or the whole TAP system. Quite the opposite, participants might not only be unable to take advantage of either richer rule structures or richer system descriptions but they might also be confused by them. According to the idea that "easy is (almost always) better" when dealing with non-programmers (cf., the approach underlying IFTTT), we might even find an advantage for either the IF-THEN or the *Descriptive* condition.

3.1 Materials

The study material—tutorial and scenarios—was prepared in Italian. In the rules that accompanied the scenarios, we used the Italian conjunctions "SE," "QUANDO," and "MENTRE" for the English "IF," "WHEN," and "WHILE," respectively.

The tutorial was realized as a short, written document illustrating a "smart house" as a home environment equipped with a set of electronic devices (automatic doors and windows, automatic lights, weather station, sensors of movements and presence). Management of a smart home is an interesting application for TAP [13, 58] and it is easy to communicate.

The tutorial briefly explained how these devices could be used as sensors and actuators by providing a few examples. Then, an additional short example involving a kettle was presented to illustrate the difference between events and states. A graphical representation supported the description of this example (Fig. 1). The rules were presented to participants both in the "IF-THEN" and the "WHEN-WHILE-THEN" forms. The last part of the tutorial was provided in two different versions: one aimed at communicating a *Computational* model for the working of the rules and another one aimed at communicating a *Descriptive* model. The whole tutorial (both the Italian and English versions) is included in Appendix A.

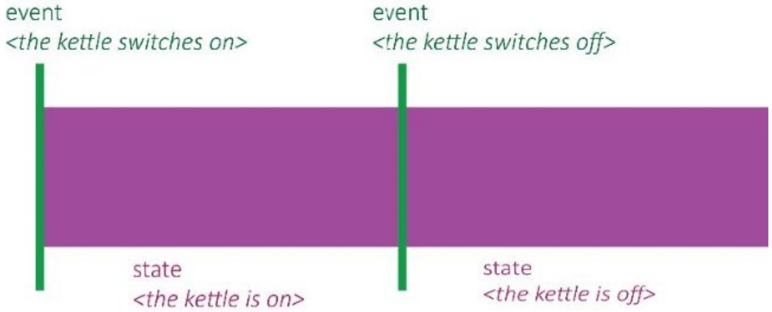


Fig. 1. Example used in the tutorial to illustrate the difference between events and states.

A short questionnaire (Appendix B) with six multiple-choice questions was used to test participants' comprehension of the information presented in the tutorial. There were four response alternatives for each question. Five questions assessed participants' understanding of the difference between states and events (i.e., these five questions were the same for participants of the *Computational* and *Descriptive* groups), while the last question assessed their understanding of how the automatic system described in the tutorial worked (i.e., the last question was specific to the group to which participant belonged).

The scenarios were based on the fictitious domain of controlling a "smart house." Each scenario presented the description of a given desired situation and one or two rules that supposedly realized that situation (i.e., the situation was presented as the desired outcome, which (possibly) resulted from the execution of the rules). There were two different versions of each scenario and associated rules: the rules might be in either the IF-THEN or WHEN-WHILE-THEN format. The two formats were counterbalanced between participants (each participant saw each rule only in one format).

An example of a scenario is the following: "You want the windows of your house to close if it rains, but when it stops, they should be opened if somebody is at home." The two rules in this scenario are the following: WHEN [it starts raining] THEN [close all the windows] (in the IF form: IF [it starts raining] THEN [close all the windows]), and WHEN [it stops raining] WHILE [somebody is at home] THEN [open all the windows] (in the IF form: IF [it stops raining] and [somebody is at home] THEN [open all the windows]).

There was a multiple-choice question for each scenario that tested participants' understanding of the rules' behavior (i.e., whether the rules realized the desired situation and, if not, which outcome was expected to occur). We presented four alternative answers for each question, one of which was the correct one (i.e., it described the effect of the rules in that situation). For the example above, the question was: "You are at home, and the windows are open; it starts raining; what does it happen to the windows?" The four alternatives were (1) "the windows will remain open," (2) "the windows will close" (i.e., the correct alternative), (3) "the windows will close, and then they will immediately open again," and (4) "it cannot be determined because the answer depends on other factors."

Overall, we prepared six scenarios, four were "unbuggy" (the rules realized the desired situation) and two were "buggy" (the rules did not realize the desired situation). The two "buggy" scenarios were based on either the "Infinite loop" bug (one rule triggered another rule, which then triggered the first, ad infinitum) or the "repeated triggering" bug (a rule repeatedly triggered because a state remained true even after the execution of the rule's action) described by Brackenbury et al. [26]. Each participant thus saw three scenarios with rules having the WHEN-WHILE-THEN format and three scenarios with rules having the IF-THEN format. Among scenarios with rules of either format, there was one buggy scenario. The six scenarios are reported in Appendix C.

Participants were also given a table describing all the possible devices mentioned in the experimental scenarios, including related events, states, and actions. Crucially, events, states, and actions were printed in different colors (i.e., we used green, purple, and blue characters for events, states, and actions, respectively). The same colors were used in the scenarios when the rules were described. Accordingly, an inappropriate interpretation of the behavior of the rules (i.e., the choice of an incorrect answer in the questions assessing rules' understanding) could not be attributed to the participants being confused about whether a given occurrence was an event or a state.

3.2 Participants and Procedure

Thirty students of the Department of Psychology and Cognitive Science of the University of Trento (28 females; mean age, 20.13±2.80 years) volunteered to participate in Study 1. The condition for inclusion was no computer programming experience and good Italian fluency. There was no compensation for participation in the study.

An a-priori power analysis performed with G*Power 3.1 [59] showed that, for a mixed Analysis of Variance (ANOVA) with $\alpha=0.05$, this number of participants resulted in a power of 0.75 in detecting a

2×2 interaction with a medium effect size (Cohen's f=0.25; i.e., the a-priori power value was very close to the optimum value of 0.8).

Participants performed the task at home, using their own personal computer equipment. They were instructed to find a quiet room where they could perform the task without being disturbed. Each participant received a personal link to the task that was assigned to him/her and performed it individually. Half of the participants were randomly assigned to the *Computational* condition while the other half were assigned to the *Descriptive* condition. The two possible types of scenarios (i.e., with IF-THEN or WHEN-WHILE-THEN rules) were counterbalanced between participants. Each participant saw the scenarios (of either type) in a randomized order. Participants were required to start and finish the task in a single session, but no time limits were imposed.

Participants first read either the *Computational* or *Descriptive* tutorial (according to the between-subject condition assigned to them) and then responded to the questionnaire assessing their comprehension of the tutorial. After completing this questionnaire, they read the scenarios and, for each scenario, they answered the question assessing the understanding of the rules' behavior. Participants also had to report their confidence in their answers on a 5-point Likert scale.

The tutorial and table with the description of the devices (with associated events, states, and actions) remained available for consultation for the whole duration of the experimental session.

3.3 Results

3.3.1 Tutorial comprehension questionnaire

Participants demonstrated a good understanding of the notion of event and a reasonably good understanding of the notion of state: all participants correctly answered that an event is something that "happens in the house in a given moment" and only two of them did not correctly answer that a state is "something that has a duration and can be true or false." Most participants also showed that they had quite well understood the structure of the rules: all participants chose at least one of the two possible correct responses when asked about which word, among WHEN, WHILE, IF, and THEN, should precede either events or states. In the question about events, 24 participants chose both correct response alternatives without selecting any incorrect alternative, while, in the question about states, only 13 participants gave a fully correct response. Nevertheless, most of the partial errors in both these questions consisted of either choosing only one correct response or choosing THEN in addition to the two correct alternatives. Indeed, only five participants incorrectly chose either WHEN in the case of a state or WHILE in the case of an event. In the question assessing the comprehension of the part of the tutorial that differed between the *Computational* and *Descriptive* groups (i.e., the question that was different for participants of the two groups) only six participants (four from the *Descriptive* group) made a mistake. Chi-square analyses revealed that for none of these questions there was a significant difference in the correct versus incorrect response distributions between the two groups (all $\chi^2 \leq 0.83$, all $p \geq 0.36$). Overall, therefore, the two groups did not differ in the comprehension of the tutorial.

On the question concerning the conditions that need to be met for a rule to be activated, most participants responded correctly: only four participants made a mistake. All these participants were from the Computational group. In this question, the difference between the two groups was significant ($\chi^2=4.615$, $p=0.032$).

3.3.2 Scenarios and related questions

More participants from the *Computational* than from the *Descriptive* group responded correctly in the buggy scenarios, both when the rules were presented in the WHEN-WHILE-THEN format (10 vs. 7) and when they were presented in the IF-THEN format (7 vs. 2), even if only this last difference was statistically significant ($\chi^2=3.968$, $p=0.046$). The number of participants from the *Computational* group who responded incorrectly to the questions of the buggy scenarios was still quite high (five and eight participants in the case of WHEN-WHILE-THEN and IF-THEN formats, respectively). Nevertheless, it is worth noting that among these participants there were the two participants who had also responded incorrectly to the question assessing the comprehension of the cyclical mechanism of the TAP system (i.e., the part of the tutorial whose understanding was critical to detect the bugs in the buggy scenarios).

Participants' proportions of correct responses and mean Likert confidence rating scores from all the scenarios (i.e., both buggy and unbuggy) are shown in Table 1. These data were entered into two ANOVAs with one within-participants factor and one between-participants factor: rule structure (IF-THEN vs. WHEN-WHILE-THEN rules) and depiction of the TAP system in the tutorial (*Computational* vs. *Descriptive*). The accuracy ANOVA showed a significant main effect of the rule structure: proportions of correct responses in the WHEN-WHILE-THEN condition were higher than those in the IF-THEN condition (0.79 vs. 0.64; $F_{(1,28)}=9.10$; $p=0.005$). Participants from the *Computational* group responded more accurately than those from the *Descriptive* group (0.76 vs. 0.68), but the effect of the system's depiction was not statistically significant $F_{(1,28)}=1.32$; $p=0.26$). The interaction between the two factors was also not significant ($F_{(1,28)}=0.05$; $p=0.82$).

The confidence rating scores showed the same trend as accuracy data. However, in the confidence ANOVA, there were no significant main effects or interactions (all $F_{(1,28)}\leq 1.44$; all $p\geq 0.24$).

As shown in Table 1, the average confidence ratings were, in general, relatively high.

Table 1. Scenarios and related questions

Depiction of the TAP system	Format	Proportions of correct responses	Likert confidence rating scores
Computational	WHEN-WHILE-THEN	0.82±0.25	4.13±0.61
	IF-THEN	0.69±0.23	4.02±0.90
Descriptive	WHEN-WHILE-THEN	0.76±0.20	4.00±0.71
	IF-THEN	0.60±0.23	3.80±0.76

Values are presented as mean±standard deviation of proportions of correct responses and Likert confidence rating scores (range 1–5) of participants receiving a *Computational* vs. *Descriptive* depiction of the TAP system as a function of the format of the rule (WHEN/WHILE-THEN vs. IF-THEN).

3.4 Discussion

Results of Study 1 provide evidence that a structured format of the rules, compared to a simpler (less structured) one, can effectively help users better understand the meaning of rules in determining a given scenario: in both groups of participants, the WHEN-WHILE-THEN format led to more accurate performances than the IF-THEN format.

This study also provides some evidence for an impact of how the rule system is described on the comprehension of the rule behavior. A description of the system that specified the cyclical nature of the rules' evaluation and activation mechanism proved beneficial, compared to a simpler description of the system, at least when participants dealt with buggy scenarios and rules with the most difficult (IF-THEN) format. Presumably, a description of the cyclical mechanism allowed for a more efficient mental model of how the system works, which helped participants understand when the described scenario was buggy (i.e., the rules would not have led to the desired outcome). No significant effect was found when the analyses also involved the scenarios in which the knowledge about this cyclical mechanism was less critical (i.e., the scenarios that did not involve a repeated application of the rules).

Results of this study extend previous research (specifically, Brackenbury et al [26] and Huang and Cakmak [25]), providing evidence for the usefulness of both a language-based nudging strategy and an informing strategy. In fact, the impact of the informing strategy and a computational mental model on performance may have been underestimated in the present study. Even though the depiction of the system in the *Descriptive* condition did not contain an explicit description of the cyclical mechanism, it contains a clear and explicit description of the rule triggering conditions. Indeed, in the preliminary comprehension questionnaire, the question about the conditions that need to be met for a rule to be activated was responded to more accurately by participants from the *Descriptive* group than from the *Computational* group. This greater accuracy may be traced back to the form of this question (i.e., lexical choices and how sentences were structured), resembling very closely how the triggering conditions were explained in the *Descriptive* tutorial. Such a greater accuracy may therefore simply result from an easier matching between the information provided in the tutorial and that required in the question. Crucially, however, it may also reflect a truly better understanding of the triggering conditions and the effect of their composition (e.g., the fact that all the conditions need to be satisfied for the action part of the rule to be executed). It is worth noting that, in the *Descriptive* tutorial, the description of these conditions was the critical (and, basically, the only relevant) point, beyond the distinction between event and states. Accordingly, participants of the *Descriptive* group mostly needed to focus on this aspect. In contrast, the *Computational* tutorial focused on the cyclical mechanism of rule evaluation and activation. In this tutorial, the consequences of the fulfillment of the rule triggering conditions (or of the lack of it) on rule activation were not explicitly described and can only be inferred.

Therefore, the slight difference found between the *Descriptive* and *Computational* groups in the questions about the scenarios may result from a true advantage given by the *Computational* model for the understanding of the rule evaluation and activation mechanism, which however was partially counterbalanced by a better comprehension of the triggering conditions in the *Descriptive* condition. Furthermore, as noted above, in Study 1, only two scenarios involved the repetition of the rules and thus critically dependent on the knowledge about the cyclical mechanism of rule evaluation and activation. That too may have prevented the *Computational* description's advantage from fully emerging.

Finally, it is worth noting that knowledge of this mechanism is only a part of users' mental model of a TAP system (i.e., the conceptual or long term memory model of the system—in the present case, the system ruling the automated smart home) and, in turn, the system mental model is only one of the sources of information on which users rely when they create a mental representation of a specific instance of interaction with the system in working memory [60]. Other sources of information are the knowledge of the structure of the rules used to customize the system's operations, the properties of the language used to express these rules, the representation of events, states and actions and their relations. In Study 1, all these different aspects likely interacted with each other and each of them may have either enhanced or reduced the effect of the other on participants' performances. In order to better understand such mental representations of user-system interactions, we planned a second study.

4. Study 2: A Qualitative Investigation of Participants' Verbal Reports in a (Fictitious) TAP Task

The main objective of the second study was to get a better understanding of naïve users' mental representations of interactions with a TAP system (i.e., the automated smart home presented in Study 1).

Following other studies aimed at eliciting mental models of technologies [e.g., 37, 61–63], we decided to employ a qualitative approach with an interpretative stance [64]. Participants were interviewed while performing the same task as that administered in Study 1, and verbal reports were collected and analyzed. To analyze interview data, we used the so-called thematic analysis [65–67]. Thematic analysis is a very common type of analysis in qualitative research. It is largely used in social and health research, and it has been also successfully employed in HCI and software engineering (e.g. [68, 69]). Being a qualitative type of analysis, this approach does not strive to account for quantitative differences in the collected data but rather aims to explore the "*why*" of observed phenomena. Specifically, this type of analysis strives to identify patterns of topics, concepts, meanings, and ideas that come up repeatedly in the interview data. For these reasons, it was the optimal choice to explore the actual mental models created by non-expert users while interacting with the (fictitious) smart-home system based on TA rules that we proposed in the present research work, that is, it was the optimal tool to analyze the thoughts, observations, and remarks that participants freely expressed while they were performing the task and trying to predict the outcomes of these rules.

4.1 Materials, Participants, and Procedure

Study 2 used the same scenarios and rules as those in the previous study. The procedure was adapted to a qualitative study. Participants were not alone while they were performing the task, but a facilitator was present via video conference. After a short introduction of the smart home context orally presented by the facilitator, participants read the scenarios in individual sessions. The tutorial was purposefully not used in this study to let the participants freely think about the different concepts involved in the scenarios and related questions. Only the *Descriptive* depiction of the system was presented. Half of the participants were presented with the rules in the IF-THEN format and the other half were presented with the rules in the WHEN-WHILE-THEN format.

Participants were asked to answer the scenarios' questions. Besides choosing one of the four alternatives, they had to verbally explain their understanding of the rules. Furthermore, participants were prompted by the facilitator to discuss their understanding of the distinction between states and events.

A total of 14 subjects (7 males and 7 females from 20 to 40 years old) participated in the study. They had been recruited using a snowball procedure starting from personal acquaintances. The inclusion criteria were the lack of any computational experience and no knowledge of programming languages. All participants were native Russian speakers except for P10, a native Portuguese speaker, and all spoke (fluent) English as a second language. Each participant was interviewed individually for about 30 to 40 minutes (8 hours overall). The interviews were conducted in English; they were audio-recorded and transcribed for analysis.

4.2 Results

The data (participants' verbal reports) were analyzed following the tenets of thematic analysis [65–67]. In thematic analyses, participants' verbal reports are systematically analyzed in order to detect common topics (called "codes"). Specifically, the coding of the verbal reports is done iteratively, initially with an inductive, data-driven, approach (i.e., data are first coded without trying to fit the coding process into a preexisting coding frame and thus without focusing on the specific aims of the questions that were asked to participants). Then, the codes are grouped into clusters that are called "themes" and represent theoretical dimensions that can explain the data. Eventually, the codes are retrospectively reconsidered with a deductive approach [67].

The analysis of our interviews was performed by two independent evaluators who compared the outputs of their analyses and converged on a limited number of codes and themes. In this analysis, six codes were identified (i.e., events as actions, states as longer activities. states as situations, events as the starting and ending points of states, states as movements, states, and events as a function of sensors), which were related to three themes (i.e., distinction between events and states, temporality of events, actions as an overarching category) (Fig. 2).

Many participants focused on actions as a conceptual primitive for the notion of event. For example, "When I come home, it's an event because it is an action." (P9), "It's just one moment to step into the house." (P13), "It's an action. Someone must leave the house." (P5), and "When you enter, it's the action." (P8).

Sometimes, states are also conceptualized as activities that take place in the house, but they are seen as having a different duration from that of events. For example, "When you are entering the backyard, it's a very quick action. When you are inside the backyard, that's a longer action." (P8) and "What if I just went to my backyard to take my ...to take a tool that I need to work inside the house. I didn't stay in the backyard for at least one minute let's say." (P8).

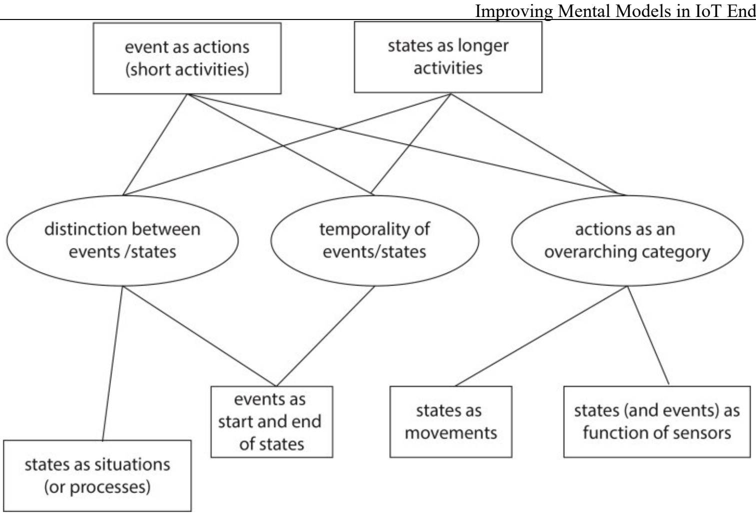


Fig. 2. Themes and codes that emerged from the thematic analysis of the participants' interviews. Codes are represented as squares and themes as ovals; a link between a code and a theme identifies the code as belonging to that theme.

Events were often described as something that happens "in a moment." For example, "Then someone leaves the house … I mean the moment people go away." (P7), "And entering the yard it's like opening the door and stepping into it. Time durations...in terms is like a very short moment." (P4), "Being in the backyard, it takes time, but entering, it takes just seconds, some seconds, I think." (P11), and "Couple of seconds to enter the backyard. It's a moment nobody is at home." (P15).

In contrast, a state was often described as something that lasts for some time or happens in a time range. For example, "And you are at home, you kind of like ehm... staying at home for some period of time." (P13), "I guess, we can put this... timing from which... for instance, from 9 in the evening to 6 a.m. It should switch the lights on." (P13), "Someone comes home, it's a period of time." (P5), "And the process is staying at home. For me it's like a longer process. From the time you enter till the time you exit. Everything in between is you are staying at home." (P13), and "But it is raining, it can be a long time." (P14).

In some cases, states were conceptualized as situations (e.g., "[It] is a condition, it is already a situation." (P8)) while in other as processes (e.g., "I don't know, may be like actions are enter and exit and process is staying at the home. For me it's like a longer process." (P13) and "If someone there like physically doing something, that is being in the yard." (P4)).

Several participants seem to be somehow confused by the relation that events and states have with the actual sensors: "Ok, somebody is at home also it feels by sensors, people are moving..." (P4), "It's almost the same as fridge, for example, you open the door of the fridge, and it knows that you need the light. There is a sensor, so it turns on. When you close the door in the fridge, it turns off. Rights, so the same for the house." (P3), and "When I passed the sensor, my status changed." (P14). In at least one case, this confusion was clearly elaborated: "I am not sure whether I understand the difference between it's raining and it starts raining because what triggers me to understand that it is raining outside, this is like a sensor that feels the water." (P14).

Nevertheless, some verbal reports show that the relation between events and states was somehow understood (more or less clearly). For example, "I would say it is like a status of a user, so there is a user, and my status is out, so I am not in the backyard." (P14), "[…] Everything in between is you are staying at home." (P13), "When I walk to the backyard, it starts and it finishes when I walked out of the backyard." (P12). Verbal reports suggesting that the critical distinction was understood were more frequent in the case of rules containing both WHEN and WHILE triggering conditions.

In general, participants exposed to the WHEN-WHILE-THEN rule format seemed to have better understood the difference between events and states, including the temporality issue.

4.3 Discussion

Results of this study reveal some critical aspects of users' mental representations of the automated smart home and the interaction with this system. Many participants seemed to intuitively understand the distinction between events and states, albeit not always clearly articulated. Participants referred to events and states with general terms like status, activity, and action. Interestingly, participants who seemed to have understood that there was a difference between the event- and state-parts of the rules' triggering conditions tended to use a specific term for the event-part (i.e., a term often related to something that is deliberately performed by the person in the smart house) and other, less consistent, terms or periphrases for the state-part.

As noted above, the notion of "action" (or "activity") emerges as an overarching category and temporal aspects seem to have driven the distinction between events and states: short actions are events while longer activities are states. The use of the WHEN-WHILE-THEN format for the rules seems to have helped participants understand this critical distinction.

The meaning attributed by participants to both events and states (i.e., user's action/activity) might have been a source of confusion, as events and states appeared in the triggering part of the rules, before "THEN," while the term "action" was used instead by the facilitator to indicate the operations performed by the system (as it is generally used in all TAP systems) which, in the rules, appeared after "THEN."

A final aspect worth noticing is that the description of sensors seems to have been more confusing than helpful. This aspect should be considered when giving instructions about a specific TAP system to naïve users; the way sensors are introduced may bias the mental model users derive on how the system works.

Results of this study bring to light the limitations of the language-based nudging strategy that we propose: even if two different temporal conjunctions are used to introduce events and states, without a detailed explanation of the difference between these two types of occurrences, participants often still have some difficulties in discriminating between them. In this respect, these results extend those of Huang and Cakmak [25] and Mattioli and Paterno [32] by providing additional evidence for both the pervasiveness and negative impact of this critical difficulty among naïve users.

5. General Discussion

Our main goal was to evaluate two different strategies to improve the mental models of naïve users for programming IoT environments. We propose a nudging strategy (i.e., changing the usual structure of the rules and using two different temporal conjunctions to introduce the event- and state-parts of the triggering conditions) and an informing strategy (i.e., providing an explicit description of the cyclical nature of the rule mechanism). Our work has several limitations, including the impromptu nature of the tasks, the small number of participants involved, the fact that they were all young adults (mostly students), thus not representative of all possible users of TAP systems, and that they did not really "test" a TAP system in real life, but just discussed scenarios and chose an answer among four options, based on their comprehension of the scenarios. Nevertheless, we believe that our mixed approach, experimental and exploratory, allowed us to draw two meaningful lessons to inform the evolution of EUD systems for IoT.

The first lesson is that the two strategies do work. In particular, the WHEN-WHILE-THEN rule structure appears to be truly helpful. By using it, we can exploit the implicit linguistic knowledge about the difference between states and events. Properly designed graphical interfaces may use this structure to guide users in defining rules [25, 26, 56]. This proposal is consistent with the idea that natural language can be effectively exploited to assist TAP and with those approaches to programming based on natural language instructions (cf., [47, 52, 53]). Both Studies 1 and 2 indeed provide evidence that the use of the temporal conjunctions WHEN and WHILE may help users interpret the rules correctly when these rules involve events and states.

Nevertheless, the second study also suggests that, without detailed explanations using specific and different terms for the two types of occurrences, some users may still not be able to rationalize this difference, even though they can discuss the difference between longer and shorter events. Multi-clause rules, involving both an event and a state introduced by WHEN and WHILE, respectively, seem to have driven a better understanding of the distinction between the two types of occurrences.

These data are consistent with linguistic and psycholinguistic evidence on the comprehension of temporal sentences. As observed by de Vega et al. [33], when temporal sentences involve two simultaneous occurrences, one of them tends to be interpreted as the main one while the other occurrence is seen as the "ground" (i.e., the context) where the main event occurs. These authors found that (1) the occurrence that takes more time is usually seen as the ground and (2) sentences in which the longer (ground) occurrence is introduced by WHILE are judged as more acceptable and sensible than sentences in which this occurrence is introduced by WHEN. They conclude that WHILE is the temporal conjunction that people usually see as introducing prolonged occurrences (the "states" in our terminology) that act as the context for other occurrences. Accordingly, when participants are presented with two-condition rules in which the event and state conditions are introduced by WHEN and WHILE, respectively, they may more easily identify the context (the state) in which something (the event) is happening (i.e., participants may more "naturally" understand the semantics of the state and event contained in the rule, thus better understanding the distinction between them and the meaning of the whole rule). It is worth noticing, however, that WHEN and WHILE do not simply act as cues able to help participants distinguish the two different parts of the rules' triggering conditions. Indeed, in Study 1 the event and state parts were written in different colors (in both the WHEN-WHILE-THEN and IF-THEN rules) and a list of all the possible events and states of the automated smart home was always available to participants. No confusion about whether a given occurrence was a state or event could occur. Nevertheless, the use of the WHEN-WHILE-THEN format proved to have a beneficial effect on performance.

When this format was used, there is no additional effect of the system description provided to participants: the *Computational* depiction of the system helped participants detect buggy scenarios but the advantage of the *Computational* group over the *Descriptive* group was only significant when the IF-THEN rules were considered. Based on these findings, we may conclude that, in such buggy conditions, either an appropriate rule format or a proper system description is enough to help people understand that rules do not work as expected. However, as noted above, by using the difference between the performances of the two groups, we may have underestimated the effect of the informing strategy: the *Computational* depiction of the system may have been more informative as to the cyclical mechanism of rules' evaluation and activation, but the *Descriptive* depiction may have been more informative as to the conditions required to trigger the rules. Accordingly, the system mental model of participants from the *Computational* group might be more appropriate regarding the former aspect but less appropriate with regard to the latter.

The second lesson drawn from Studies 1 and 2 is that, in order to design effective interfaces, we need to know how people actually learn EUD and how they can develop appropriate mental models of the system with which they interact. When dealing with naïve users (i.e., the main target group in EUD), the complexity of the programming constructs is not the whole story and making them simpler is not the only solution to pursue. Results of the second study suggest that the understanding of the rules and of the task itself may be compromised by participants' previous naïve assumptions on what sensors are, as well as by the confusion between events and states and between the user's actions, which are often involved in the rule triggering conditions (i.e., the parts of the rule introduced by IF, WHEN or WHILE), and the system's operations (i.e., the action part of the rule that is introduced by THEN). In fact, in participants' verbal reports, the notion of "action" appears to be an overarching category that includes both the operations performed by the system and the description of the situation in which they are performed. The use of different terms, more specifically linked to the notion of "action" and "operation", to introduce the action part of the rule (e.g., DO instead of THEN; cf., [56]), may be helpful to prevent such confusion.

To our knowledge, no research has been published on how people learn EUD. Although there have been a few longitudinal studies on how people control smart homes (e.g., [9, 58]), they targeted tech-savvy users and usually focused on appropriation more than learning. A better investigation of how people learn EUD might also bring new light to the timely topic of computational thinking. EUD and computational thinking are related but, to some extent, opposite concepts. The goal of EUD is basically to allow users without technical experience to program [6, 7]. Therefore, EUD promotes a kind of programming that does not heavily rely on specific computational skills. In contrast, computational thinking [70] is the kind of analytical thinking that underlies programming. How much computational thinking is needed for EUD is still an unaddressed question.

6. Conclusion

In this paper, we discuss two strategies to improve the mental models of naïve users for programming IoT environments: a nudging strategy and an informing strategy. Both studies described here provide some evidence that, when the triggering conditions of TA rules involve events and states, the rules are better interpreted if these conditions are introduced by different temporal conjunctions: WHEN (for events) and WHILE (for states). When this nudging strategy is applied, the addition of the second (informing) strategy does not provide any further significant benefit.

The second study suggests that, even when the two conjunctions are used in the rules, naïve users may still not be able to rationalize the difference between events and states. This study also emphasizes the importance of (1) the mental representations that naïve people have of how automatic systems work and (2) the lexical choices made when presenting the problems to the users.

Our research has several limitations (e.g., the small number of participants and the limited meaningfulness of the administered task; see Section 5 General Discussion) and further studies are undoubtedly needed to fully explore mental models in EUD, possibly with tasks in which participants are required to compose, rather than evaluate, TA rules in order to program their own IoT devices. However, we believe that the research work presented here highlights two crucial aspects: (1) how the task is explained (i.e., lexical choices, how the arguments are phrased, etc.) is essential and can be used to nudge users to create effective mental models of both the system and of their interactions with the system; (2) the representation of the domain and its components has also an impact on EUD mental models: users' knowledge of the domain needs to be taken into account and lexical choices need to be carefully made in order to avoid inappropriate users' mental representations. That constitutes the original contribution of our research: it supports and extends the recent literature in the field of EUD [23–28, 32], thus helping this field get closer to its ultimate goal of empowering non-expert users in a more personalized approach to IoT.

Author's Contributions

Conceptualization, BT, MZ. Funding acquisition, BT. Investigation and methodology, BT, MZ, GG. - Project administration, GG, DY. Supervision, BT, MZ. Writing of the original draft, BT, MZ. Writing of the review and editing, BT, MZ. Formal analysis, BT, MZ. Data curation, GG, DY. All the authors have proofread the final version

Funding

This research was supported by the "Progetti di Rilevante Interesse Nazionale (PRIN) 2017" Program funded by the Italian Ministry of University and Research (MUR) (No. 2017MX9T7H, EMPATHY: EMpowering People in deAling with internet of Things ecosYstems).

Competing Interests

The authors declare that they have no competing interests.

References

[1] J. H. Nord, A. Koohang, and J. Paliszkiewicz, “The Internet of Things: review and theoretical framework,” *Expert Systems with Applications*, vol. 133, pp. 97-108, 2019.

[2] C. Ardito, P. Buono, G. Desolda, and M. Matera, “From smart objects to smart experiences: an end-user development approach,” *International Journal of Human-Computer Studies*, vol. 114, pp. 51-68, 2018.

[3] H. Lieberman, F. Paterno, M. Klann, and V. Wulf, “End-user development: an emerging paradigm,” in *End User Development*. Dordrecht, Netherlands: Springer, 2006, pp. 1-8.

[4] F. Paterno and C. Santoro, “End-user development for personalizing applications, things, and robots,” *International Journal of Human-Computer Studies*, vol. 131, pp. 120-130, 2019.

[5] D. J. Langley, J. van Doorn, I. C. Ng, S. Stieglitz, A. Lazovik, and A. Boonstra, “The Internet of everything: smart things and their impact on business models,” *Journal of Business Research*, vol. 122, pp. 853-863, 2021.

[6] G. Fischer, E. Giaccardi, Y. Ye, A. G. Sutcliffe, and N. Mehandjiev, “Meta-design: a manifesto for end-user development,” *Communications of the ACM*, vol. 47, no. 9, pp. 33-37, 2004.

[7] F. Paterno, “End user development: Survey of an emerging field for empowering people,” *International Scholarly Research Notices*, vol. 2013, article no. 532659, 2013. https://doi.org/10.1155/2013/532659

[8] IFTT Web-Based Service [Online]. Available: https://ifttt.com/.

[9] B. Ur, M. Pak Yong Ho, S. Brawner, J. Lee, S. Mennicken, N. Picard, D. Schulze, and M. L. Littman, “Trigger-action programming in the wild: An analysis of 200,000 IFTTT recipes,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose, CA, 2016, pp. 3227-3231.

[10] Amazon's Alexa [Online]. Available: https://developer.amazon.com/en-US/alexa.

[11] J. F. Pane and B. A. Myers, “Studying the language and structure in non-programmers' solutions to programming problems,” *International Journal of Human-Computer Studies*, vol. 54, no. 2, pp. 237-264, 2001.

[12] A. K. Dey, T. Sohn, S. Streng, and J. Kodama, “iCAP: interactive prototyping of context-aware applications,” in *Pervasive Computing*. Heidelberg, Germany: Springer, 2006, pp. 254-271.

[13] B. Ur, E. McManus, M. Pak Yong Ho, and M. L. Littman, “Practical trigger-action programming in the smart home,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Toronto, Canada, 2014, pp. 803-812.

[14] S. Davidoff, M. K. Lee, C. Yiu, J. Zimmerman, and A. K. Dey, “Principles of smart home control,” in *UbiComp 2006: Ubiquitous Computing*. Heidelberg, Germany: Springer, 2006, pp. 19-34.

[15] G. Ghiani, M. Manca, F. Paterno, and C. Santoro, “Personalization of context-dependent applications through trigger-action rules,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 24, no. 2, article no. 14, 2017. https://doi.org/10.1145/3057861

[16] SmartThings [Online]. Available: https://www.smartthings.com/.

[17] SharpTools [Online]. Available: https://sharptools.io/.

[18] N. W. Paton, *Active Rules in Database Systems*. New York, NY: Springer Science & Business Media, 1998.

[19] C. W. Tan and A. Goh, “Implementing ECA rules in an active database,” *Knowledge-Based Systems*, vol. 12, no. 4, pp. 137-144, 1999.

[20] F. Casati, S. Castano, M. Fugini, I. Mirbel, and B. Pernici, “Using patterns to design rules in workflows,” *IEEE Transactions on Software Engineering*, vol. 26, no. 8, pp. 760-785, 2000.

[21] J. Bae, H. Bae, S. H. Kang, and Y. Kim, “Automatic control of workflow processes using ECA rules,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 8, pp. 1010-1023, 2004.

[22] U. Dayal, M. Hsu, and R. Ladin, “Organizing long-running activities with triggers and transactions,” in *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, Atlantic City, NJ, 1990, pp. 204-214.

[23] J. Brich, M. Walch, M. Rietzler, M. Weber, and F. Schaub, “Exploring end user programming needs in home automation,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 24, no. 2, article no. 11, 2017. https://doi.org/10.1145/3057858

[24] P. Frohlich, M. Baldauf, P. Palanque, V. Roto, T. Meneweger, M. Tscheligi, Z. M. Becerra, and F. Paterno, “Automation experience across domains: designing for intelligibility, interventions, interplay and integrity,” in *Proceedings of CHI EA 2020: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, 2020, pp. 1-8.

[25] J. Huang and M. Cakmak, “Supporting mental model accuracy in trigger-action programming,” in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Osaka, Japan, 2005, pp. 215-225.

[26] W. Brackenbury, A. Deora, J. Ritchey, J. Vallee, W. He, G. Wang, M. L. Littman, and B. Ur, “How users interpret bugs in trigger-action programming,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, UK, 2019, pp. 1-12.

[27] K. N. Truong, E. M. Huang, and G. D. Abowd, “CAMP: a magnetic poetry interface for end-user programming of capture applications for the home,” in *UbiComp 2004: Ubiquitous Computing*. Heidelberg, Germany: Springer, 2004, pp. 143-160.

[28] G. Desolda, C. Ardito, and M. Matera, “Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools,” *ACM Transactions on Computer-Human Interaction*, vol. 24, no. 2, article no. 12, 2017. https://doi.org/10.1145/3057859

[29] R. Casati and A. C. Varzi, “Event concepts,” in *Understanding Events: From Perception to Action*. New York, NY: Oxford University Press, 2008, p. 31-53.

[30] F. Pianesi and A. C. Varzi, “Events and event talk: an introduction,” in *Speaking of Events*. New York, NY: Oxford University Press, 2000, p. 3-47.

[31] R. Montague, “On the nature of certain philosophical entities,” *The Monist*, vol. 53, no. 2, pp. 159-194, 1969.

[32] A. Mattioli and F. Paterno, “A visual environment for end-user creation of IoT customization rules with recommendation support,” in *Proceedings of the International Conference on Advanced Visual Interfaces*, Salerno, Italy, 2020, pp. 1-5.

[33] M. de Vega, M. Rinck, J. M. Diaz, and I. Leon, “Figure and ground in temporal sentences: the role of the adverbs when and while,” *Discourse Processes*, vol. 43, no. 1, pp. 1-23, 2007.

[34] D. Norman, *The Design of Everyday Things: Revised and Expanded Edition*. New York, NY: Basic Books, 2013.

[35] D. Gentner and A. L. Stevens, *Mental Models*. Hillsdale, NJ: Erlbaum, 1983.

[36] S. J. Payne, “Mental models in human-computer interaction,” in *The Human-Computer Interaction Handbook*, 2nd ed. Boca Raton, FL: CRC Press, 2007, pp. 89-102.

[37] F. G. Halasz and T. P. Moran, “Mental models and problem solving in using a calculator,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Boston, MA, 1983, pp. 212-216.

[38] M. Babar, M. S. Khan, U. Habib, B. Shah, F. Ali, and D. Song, “Scalable edge computing for IoT and multimedia applications using machine learning,” *Human-centric Computing and Information Sciences*, vol. 11, article no. 41, 2021. https://doi.org/10.22967/HCIS.2021.11.041

[39] M. Malekshahi Rad, A. M. Rahmani, A. Sahafi, and N. Nasih Qader, “Social Internet of Things: vision, challenges, and trends,” *Human-centric Computing and Information Sciences*, vol. 10, article no. 52, 2020. https://doi.org/10.1186/s13673-020-00254-6

[40] S. E. Lee, M. Choi, and S. Kim, “How and what to study about IoT: research trends and future directions from the perspective of social science,” *Telecommunications Policy*, vol. 41, no. 10, pp. 1056-1067, 2017.

[41] D. Ferraris, C. Fernandez-Gago, and J. Lopez, “A model-driven approach to ensure trust in the IoT,” *Human-centric Computing and Information Sciences*, vol. 10, article no. 50, 2020. https://doi.org/10.1186/s13673-020-00257-3

[42] R. Gennari, M. Matera, A. Melonio, M. Rizvi, and E. Roumelioti, “The evolution of a toolkit for smart-thing design with children through action research,” *International Journal of Child-Computer Interaction*, vol. 31, article no. 100359, 2022. https://doi.org/10.1016/j.ijcci.2021.100359

[43] N. Batalas, I. Lykourentzou, V. J. Khan, and P. Markopoulos, “Reconsidering end-user development definitions,” in *End User Development*. Cham, Switzerland: Springer, 2021, pp. 19-35.

[44] T. Szydlo, R. Brzoza-Woch, J. Sendorek, M. Windak, and C. Gniady, “Flow-based programming for IoT leveraging fog computing,” in *Proceedings of 2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Poznan, Poland, 2017, pp. 74-79.

[45] F. Corno, L. De Russis, and A. M. Roffarello, “From users’ intentions to IF-THEN rules in the Internet of Things,” *ACM Transactions on Information Systems*, vol. 39, no. 4, article no. 53, 2021. https://doi.org/10.1145/3447264

[46] P. A. Akiki, P. A. Akiki, A. K. Bandara, and Y. Yu, “EUD-MARS: end-user development of model-driven adaptive robotics software systems,” *Science of Computer Programming*, vol. 200, article no. 102534, 2020. https://doi.org/10.1016/j.scico.2020.102534

[47] D. Fogli, L. Gargioni, G. Guida, and F. Tampalini, “A hybrid approach to user-oriented programming of collaborative robots,” *Robotics and Computer-Integrated Manufacturing*, vol. 73, article no. 102234, 2022. https://doi.org/10.1016/j.rcim.2021.102234

[48] M. Resnick, J. Maloney, A. Monroy-Hernandez, N. Rusk, E. Eastmond, K. Brennan, et al., “Scratch: programming for all,” *Communications of the ACM*, vol. 52, no. 11, pp. 60-67, 2009.

[49] N. Bak, B. M. Chang, and K. Choi, “Smart block: a visual block language and its programming environment for IoT,” *Journal of Computer Languages*, vol. 60, article no. 100999, 2020. https://doi.org/10.1016/j.cola.2020.100999

[50] M. C. Gonçalves, O. N. Lara, R. W. de Bettio, and A. P. Freire, “End-user development of smart home rules using block-based programming: a comparative usability evaluation with programmers and non-programmers,” *Behaviour & Information Technology*, vol. 40, no. 10, pp. 974-996, 2021.

[51] C. Schou, R. S. Andersen, D. Chrysostomou, S. Bogh, and O. Madsen, “Skill-based instruction of collaborative robots in industrial settings,” *Robotics and Computer-Integrated Manufacturing*, vol. 53, pp. 72-80, 2018.

[52] N. K. Lincoln and S. M. Veres, “Natural language programming of complex robotic BDI agents,” *Journal of Intelligent & Robotic Systems*, vol. 71, no. 2, pp. 211-230, 2013.

[53] J. J. Thomas, V. Suresh, M. Anas, S. Sajeev, and K. S. Sunil, “Programming with natural languages: a survey,” in *Computer Networks and Inventive Communication Technologies*. Singapore: Springer, 2022, pp. 767-779.

[54] R. Ariano, M. Manca, F. Paterno, and C. Santoro, “Smartphone-based augmented reality for end-user creation of home automations,” *Behaviour & Information Technology*, 2022. https://doi.org/10.1080/0144929X.2021.2017482

[55] R. Seiger, R. Kuhn, M. Korzetz, and U. Aßmann, “HoloFlows: modelling of processes for the Internet of Things in mixed reality,” *Software and Systems Modeling*, vol. 20, no. 5, pp. 1465-1489, 2021.

[56] G. Desolda, F. Greco, F. Guarnieri, N. Mariz, and M. Zancanaro, “SENSATION: an authoring tool to support event–state paradigm in end-user development,” in *Human-Computer Interaction – INTERACT 2021*. Cham, Switzerland: Springer, 2021, pp. 373-382.

[57] G. Gallitto, B. Treccani, and M. Zancanaro, “If when is better than if (and while might help): on the importance of influencing mental models in EUD (a pilot study),” in *Proceedings of the 1st International Workshop on Empowering People in Dealing with Internet of Things Ecosystems co-located with International Conference on Advanced Visual Interfaces (AVI)*, Ischia Island, Italy, 2020, pp. 7-11.

[58] A. Salovaara, A. Bellucci, A. Vianello, and G. Jacucci, “Programmable smart home toolkits should better address households’ social needs,” in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama, Japan, 2021, pp. 1-14.

[59] F. Faul, E. Erdfelder, A. G. Lang, and A. Buchner, “G*Power 3: a flexible statistical power analysis program for the social, behavioral, and biomedical sciences,” *Behavior Research Methods*, vol. 39, no. 2, pp. 175-191, 2007.

[60] J. J. Canas, A. Antoli, and J. F. Quesada, “The role of working memory on measuring mental models of physical systems,” *Psicológica*, vol. 22, no. 1, pp. 25-42, 2021.

[61] S. Yarosh and P. Zave, “Locked or not? mental models of IoT feature interaction,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver, CO, 2017, pp. 2993-2997.

[62] T. Ngo, J. Kunkel, and J. Ziegler, “Exploring mental models for transparent and controllable recommender systems: a qualitative study,” in *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, Genoa, Italy, 2020, pp. 183-191.

[63] K. I. Gero, Z. Ashktorab, C. Dugan, Q. Pan, J. Johnson, W. Geyer, et al., “Mental models of AI agents in a cooperative game setting,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, 2020, pp. 1-12.

[64] A. Adams, P. Lunt, and P. Cairns, “A qualititative approach to HCI research,” in *Research Methods for Human-Computer Interaction*. Cambridge, UK: Cambridge University Press, 2008, pp. 138-157.

[65] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77-101, 2006.

[66] L. S. Nowell, J. M. Norris, D. E. White, and N. J. Moules, “Thematic analysis: striving to meet the trustworthiness criteria,” *International Journal of Qualitative Methods*, vol. 16, no. 1, article no. 1609406917733847, 2017. https://doi.org/10.1177/1609406917733847

[67] J. Fereday and E. Muir-Cochrane, “Demonstrating rigor using thematic analysis: a hybrid approach of inductive and deductive coding and theme development,” *International Journal of Qualitative Methods*, vol. 5, no. 1, pp. 80-92, 2006.

[68] D. S. Cruzes and T. Dyba, “Recommended steps for thematic synthesis in software engineering,” in *Proceedings of 2011 International Symposium on Empirical Software Engineering and Measurement*, Banff, Canada, 2011, pp. 275-284.

[69] N. Cooper, T. Horne, G. R. Hayes, C. Heldreth, M. Lahav, J. Holbrook, and L. Wilcox, “A systematic review and thematic analysis of community-collaborative approaches to computing research,” in *Proceedings of CHI Conference on Human Factors in Computing Systems*, New Orleans, LA, 2022, pp. 1-18.

[70] J. M. Wing, “Computational thinking,” *Communications of the ACM*, vol. 49, no. 3, pp. 33-35, 2006.

Appendix A. The tutorial provided to participants

[This tutorial was provided to all participants of Study 1 before both the comprehension questionnaire- see Appendix B - and the experimental task- see Appendix C]

Study on the management of automatic systems by non-IT experts

In this study, we want to assess the ability of non-computer experts to control devices of complex automatic systems.

In the following, you will be presented the working of an automatic system to control devices in a smart home that can be controlled by means of rules defined by non-technical users. That system is going to be developed in the context of a research project.

You will be presented with situations, or specific scenarios, in which some of these rules are shown. For each of these situations, you will be asked to estimate what the system's behavior will be.

Your answers will help us to design the system in an easier and more comprehensible way for non-IT users.

N.B.: For this study, we ask for 40-45 minutes of your time and we ask to you to complete the whole task in one single session.

Please indicate your information:

- Gender (male, female, I prefer not to answer)
- Age [textbox]
- Education (elementary school, middle school, high school, bachelor’s degree, master's degree)

- Experience with programming languages (Yes, No - if yes, specify how long)

By going further you agree to participate in this study. The collected data are anonymous and will be stored and processed only by the persons in charge of the research. It will not be possible to trace information regarding the individual participants.

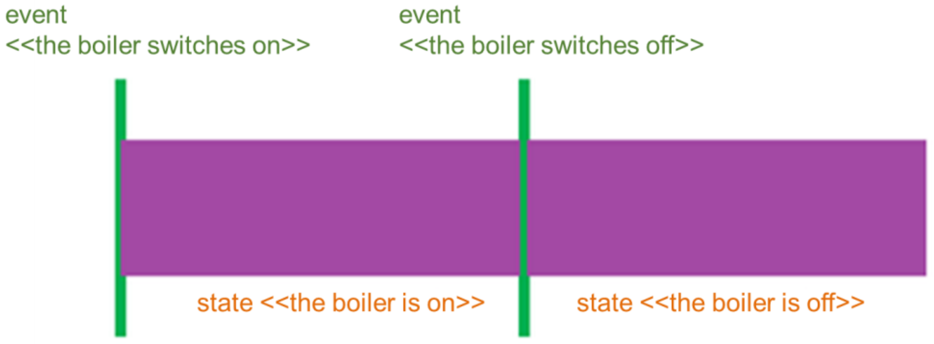
Read carefully the information you need to understand the scenarios used in this study.

Inside a highly automated house there are several electronic devices. Some of these devices collect information on the environment in which they are located, or on some of its characteristics, and are called sensors. Others, called actuators, act physically by changing the environment. A device can be, at the same time, a sensor (when it detects a situation) and an actuator (when it makes changes to the environment). For example, an automated door can determine when it is opened or closed (sensor function) and at the same time it can activate to open or close (actuator function).

Events and states

The sensors detect events and the sequence of events. Event is something that happens in the house at a specific moment (for example, "the kettle turns on") and determines a change in the state in which the device is located (for example, "The kettle is on" or "the kettle is off" depending on the sequence of events that concern it).

A state has a time duration and is processed by the system as true or false (the condition "the boiler is on" will be true as long as the boiler is actually on, that is, until it is turned off). An event, on the other hand, does not have a time duration but occurs instantaneously (for example, "the boiler switches on" is an event that occurs in a certain instant and the system is able to detect it as soon as it occurs).



The automatic system is able to detect certain events that happen in the house (depending on the available sensors) and to keep track of the change in the status of various devices.

The structure of the rules

The automatic system works according to a series of rules. The rules have the following structure:

WHEN <event> **WHILE** <state or set of states> **THEN** <action or list of actions>

<event> stands for the description of a single event; <state or set of states> stands for a single state or a list of states separated by the conjunction “AND”; <action or list of actions> stands for a single action or a list of actions that have to be executed in sequence.

It is possible to have a rule that only includes either WHEN or WHILE:

WHEN <event> **THEN** <action or list of actions>

WHILE <state or set of states> **THEN** <action or list of actions>

An another, more compact, way of defining a rule is as follows:

IF <event> **AND** <state or set of states> **THEN** <action or list of actions>

or,

IF <event> **THEN** <action or list of actions>

IF <state or set of states>**THEN** <action or list of actions>

An example for a rule is as follows:

WHEN "the alarm rings" WHILE "someone is in the bedroom" THEN "turn on the kettle" AND "turn off the bedroom air conditioner"

Or, in an equivalent way:

IF "the alarm rings" AND "someone is in the bedroom" THEN "turn on the kettle" AND "turn off the bedroom air conditioner"

How the system works [Only one of these descriptions was provided to a given participant, according to the group (descriptive or computational) to which the participant belonged]

Descriptive Description	*Computational Description*
For each rule, in case the statements on events and states in the “WHEN/WHILE” sections (or in the “IF” section for the compact form) are ALL satisfied, the automatic system performs the action or the sequence of actions described in the “THEN” section. Otherwise, the actions are not executed.	The automatic system checks each rule in the order in which they have been inserted into the system. For each rule, it first checks if the event described in the “WHEN” section (or the event in the “IF” section, in the compact form) is actually happening. In case it is happening (or if the “WHEN” section is not present), it checks if the states in the “WHILE” section (or in the IF section) correspond to the actual context of the house. If that is the case, the action or the sequence of actions described in the “THEN” section is performed (otherwise, the actions are not performed). Then, the system checks the following rule. All the rules are checked in a cycle which goes on repeatedly (the whole process takes a few milliseconds even if hundreds of rules are present).

The following table illustrates the devices related events (in green) and states (in purple) managed by the automatic system and the actions that the system can perform (in blue). You do not need to memorize the table, as it will be available during the whole study.

Device	**Functions of the device**	**Events detected by the device**	**States associated with events detected by the device**	**Actions performed by the system with reference to the device**
Atmospheric sensors	send weather information to the system send notifications to your smartphone	it starts raining it stops raining	it is raining it is not raining	send a notification to the owner's smartphone
Automated entrance door and windows	the entrance door and windows of the house can be opened or closed by the system the system receives the information if somebody rings the doorbell or not	a door opens the windows open a door closes the windows close someone rings the doorbell	a door is open the windows are open a door is closed the windows are closed	open the door/window close the door/window
Automated lights	the lights of all the rooms and of the external garden can be turned on or off by the system	the lights turn off the lights turn on	the lights are on the lights are off	turn on the lights turn off the lights
Motion sensors	detect the presence of people in the house and in the outdoor garden. it can send notifications to your smartphone	someone enters the house someone goes out of the house someone enters the garden	someone is at home nobody is at home someone is in the garden nobody is in the garden	send a notification to the owner's smartphone

Appendix B. The comprehension questionnaire

[This questionnaire was administered to participants of Study 1 after the tutorial and before the experimental task to measure the participants' understanding of the difference between states and events and of how the automatic system described in the tutorial worked]

Here are some questions regarding the section that you have just read:

An event

- it is something that happens in the house at a specific moment
- it is something that has a duration and that can be true or false
- it can be both
- None of the above answers is right

A state

- it is something that happens in the house at a specific moment
- it is something that has a duration and that can be true or false
- it can be both
- None of the above answers is right

A state is preceded by (you can mark more than one answer):

- WHEN
- WHILE
- IF
- THEN

An event is preceded by (you can mark more than one response):

- WHEN
- WHILE
- IF
- THEN

If at least one condition is not met within a rule, what happens in the THEN section of the rule?

- nothing happens
- can be one or the other
- none of the above answers is correct

[*the following question was added for the participants in the experimental condition only*]

When the system has finished checking all the rules that have been set:

- the conditions are no longer checked
- the system starts checking the rules again
- no other actions can take place
- none of the answers above is correct

[*the following question was added for the participants in the control condition only*]

The system uses the rules like this

- all conditions are checked and the actions of the rules are executed with verified conditions
- the first rule with the verified conditions is performed
- the rules are performed regardless of the initial conditions
- none of the above answers is correct

Appendix C. Scenarios

[The following scenarios composed the task administered in both Studies 1 and 2]

You will be presented with some scenarios and sets of rules. For each scenario and related rules, your task will be to choose, from a list of possible answers, the behavior that will be implemented by the system when these rules are active. The rules and the scenario associated with them contain all the information necessary to give an answer. So keep in mind a few points:

- The rules that the system uses are only those that are described together with each scenario.
- Within each rule, events will be shown in green, states in purple and actions in blue, as well as in the table.
- There are no other manipulations of the environment other than those explained in the scenario.
- The rules are always working and there are no errors in the system. However, they may not be correct, that is, they may not always lead to the goal for which they were built.
- In addition to the list of possible choices you will be presented with a box for each answer, it would be useful if you could use it to briefly explain what reasoning you made. If you don't know how to answer a question, use it to briefly describe your doubts.
- For each scenario, you will be shown a question that will ask you how confident you are of your choice on a scale that goes from "not at all safe" to "very safe". Keep in mind that "not at all sure" corresponds to a situation where you have no idea what the correct answer is.

[The infinite -loop, repeated-triggering, and window-fallacy scenarios are bugged: that is, the rules do not work as intended; the last scenarios are correct; the scenarios were presented in a randomized order]

[Infinite Loop scenarios]

1) You want the house lights to turn on when you're at home and to turn off when you're not there. To this aim, you define the following rules:

IF [the lights are on] THEN [turn off the lights] IF [someone is at home] THEN [turn on the lights]	WHILE [the lights are on] THEN [turn off the lights] WHILE [someone is at home] THEN [turn on the lights]

Enter the house. While you're inside, what will the lights be like?

- turned on
- turned off
- they will continue to switch off and on
- I don't know. The answer depends on other factors

1) You want the house windows to be open in case it does not rain and somebody is at home. While you want them to be closed in case it rains. To this aim, you define the following rules:

IF [it starts raining] THEN [close the windows] SE [it stops raining] AND [somebody is at home] THE [open the windows]	WHEN [it starts raining] THEN [close the windows] WHEN [it stops raining] WHILE [somebody is at home] THEN [open the windows]

You are at home and the windows are open. It starts raining. How will the windows be?

- Open
- Closed
- They keep opening and closing repeatedly
- I don’t know. The answer depends on other factors

[Repeated triggering scenarios]

1) To avoid forgetting the home lights on, you want them to be automatically switched off when nobody is at home. Furthermore, you want to be notified if they are switched off. To this aim, you define the following rules:

If [the lights are switching off] AND [nobody is at home] THEN [send a notification on the smartphone] IF [lights are on] AND [nobody is at home] THEN [switch the lights off]	WHEN [the lights are switching off] WHILE [nobody at home] THEN [send a notification on the smartphone] WHILE [lights are on] AND [nobody at home] THEN [switch the lights off]

You leave the house in a hurry and left the lights on, even if nobody else is in the house. After 10 minutes, how many notifications did you get on your smartphone?

- one notification only arrived
- no notifications arrived
- several notifications arrived
- I don’t know. The answer depends on other factors

2) You have to stay in the garage for a while and you want to receive a notification on your smartphone if it rains. To this aim, you define the following rule:

IF [it is raining] THEN [send a notification on the smartphone]	WHILE [it is raining] THEN [send a notification on the smartphone]

You go into the garage and, after a while, it starts raining. How many notifications did you get on your smartphone?

- one notification only arrived
- no notifications arrived

- several notifications arrived
- I don't know. The answer depends on other factors

[Time-window fallacy scenarios: for these scenarios, only the IF version was prepared because the WHEN/WHILE versions would have been too easy to recognize as bugged]

1) You live alone. Your house has a backyard and, from there, you cannot hear the doorbell ringing. To this aim, you define the following rule:

IF [somebody enters in the backyard] AND [the doorbell rings] THEN [send a notification on the smartphone]	-

It is 8 am and you go to the backyard. After 5 minutes, somebody rings the doorbell. Do you receive a notification on your smartphone?

- Yes
- No
- More than one notification is sent
- I don't know. The answer depends on other factors

2) You want the house lights to be switched on and the windows to be open every time you are back home. To this aim, you define the following rules:

IF [somebody enters the house] THEN [switch the lights on] IF [the lights are switching on] AND [somebody enters the home] ALLORA [open the windows]	-

You come back home from work. You open the door and you enter the house. At that moment, how are the windows?

- Open
- Closed
- They keep opening and closing repeatedly
- I don't know. The answer depends on other factors

[other scenarios]

1) You would like to have the door open when somebody in the house, but you want the door closed when there is nobody. For that, you defined the following rules:

IF [somebody is going out of the house] AND [nobody else at home] THEN [closed the door] IF [somebody is at home] AND [the door is closed] THEN [open the door]	WHEN [somebody is going out of the house] WHILE [nobody else at home] THEN [closed the door] WHILE [somebody is at home] AND [the door is closed] THEN [open the door]

One day, you go out of the house but there are other members of your family inside the house. Once you are out, how will the door be?

- Open
- Closed
- It keeps closing and reopening
- I don't know. The answer depends on other factors

2) You have often guests in your house and you would like that the door automatically opens if somebody rings the doorbell but only in case somebody is at home. To this aim, you define the following rules:

IF [somebody rings the doorbell] AND [the door is closed] AND [somebody at home] THEN [open the door] IF [somebody rings the doorbell] AND [nobody at home] THEN [closed the door]	WHEN [somebody rings the doorbell] WHILE [the door is closed] AND [somebody at home] THEN [open the door] WHEN [somebody rings the doorbell] WHILE [nobody at home] THEN [closed the door]

Today, you are not waiting for any guest, and you go out of the house. While you are out, somebody rings the doorbell. How will the door be?

- Open
- Closed
- It keeps closing and reopening
- I don't know. The answer depends on other factors