Expressing the Needs in Smart Home: What is the End Users' Favorite Way

XIAOHONG CHEN, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China
SHI CHEN, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China
ZHI JIN, School of Computer Science, Peking University, China
HAN BIAN, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China

The Internet of Things (IoT) has witnessed remarkable advancements, paving the way for the realization of smart homes that facilitate various user-centric scenarios. As a result, it becomes crucial to empower end users with programming capabilities to effectively express their personalized needs. Currently, the executable Trigger-Action Programming (TAP) rules have become the mainstream paradigm for IoT end-user programming. To simplify the creation of TAP rules, many studies have proposed various levels of requirement abstraction, yet the connections between these levels and TAP rules remain unclear. In this paper, we employ a trio of mixed research methods to identify the preferred way of expressing end-user requirements in practical scenarios. Subsequently, drawing from the architecture of smart homes, we distill the needs of smart homes and their environments, categorizing them into three interrelated levels of abstraction: high-level end users' intentions, medium-level intention realization plans, and low-level device scheduling logic. Based on these insights, we propose an innovative multi-level language called *SH-RDL*. Furthermore, we discuss the challenges that need to be addressed if allowing end users to express their needs using *SH-RDL*. Notably, we have conducted preliminary experiments that demonstrate how various abstraction levels within SH-RDL can be transformed into TAP rules through relevant contextual knowledge. Finally we have designed a user study to demonstrate the usability, understandability, and error-prevention benefits of the language. This will aid in the broader adoption and application of end-user programming within the IoT domain.

CCS Concepts: • **Software and its engineering → Domain specific languages**;

Additional Key Words and Phrases: IoT End-User Programming, Requirements Engineering, Smart Homes, User Intentions, Requirements Description Language

ACM Reference Format:
Xiaohong Chen, Shi Chen, Zhi Jin, and Han Bian. 2018. Expressing the Needs in Smart Home: What is the End Users' Favorite Way. In . ACM, New York, NY, USA, 24 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The advent of Internet of Things (IoT) technology has ushered in a new era where IoT-enabled systems are becoming pervasive in various facets of human society [1]. Among these systems, smart homes have gained significant attention as they seamlessly connect a multitude of household devices to offer intelligent and personalized services to users, consequently experiencing a surge in popularity [65]. By embracing such systems, users can expect a user-centered enhancement in the quality of life, characterized by heightened comfort and convenience.

As the integration of smart homes becomes increasingly prevalent in our living environment, there arises a heightened demand for personalized solutions among end users. To cater to this demand, the concept of end-user programming has been promoted, defined by Ko et al. as "a programming to achieve the result of a program primarily for personal,

Manuscript submitted to ACM

rather rather public use" [46]. It allows users to customize and orchestrate various capabilities offered by smart devices to align with their everyday requirements [29]. A prevalent paradigm for IoT end-user programming is the use of trigger-action programming (TAP) rules [31, 66]. These rules are typically formulated in the form of "IF <trigger>, THEN <action>", allowing users to establish chains of conditional statements referred to as "recipes" [56, 59]. The triggers and actions are concrete device related. For instance, a TAP rule can be defined as "IF the temperature sensor reading exceeds $31^{o}C$, THEN activate the Media air conditioner in my bedroom". The adoption of TAP rules as the primary programming paradigm is conspicuous across various IoT platforms such as IFTTT [73], Samsung's SmartThings [40], Apple's Homekit [39], and Amazon's Alexa [5].

While TAP rules could potentially satisfy most of the behaviors desired by users [74], a significant challenge faced by most smart home users lies in their struggle to effectively express their high-level expectations or intentions for the home environment using TAP rules [22, 34, 38]. TAP rules often lack a clear connection to the user's original intentions, which makes them hard to understand and hampers their reusability [22, 34]. In addition, TAP rules created by users are prone to errors, including conflicts between rules [15, 48, 67], further exacerbating the situation. To address the limitations of existing TAP rules and and simplify the creation of TAP rules, researchers have began exploring the use of higher levels of abstraction. For example, Desolda et al. introduced a new representation for IF-THEN rules using abstract smart device types [29]. Corno et al. used EUPont ontology to express both high-level requirements (functions without specifying devices) and medium-level requirements (only specifying device types) [19, 20]. Notably, Corno et al. explore potential abstraction levels through a week-long diary study, encompassing device-centric, information-centric, and people-centric abstraction [23].

The aforementioned studies have delved into numerous high-level requirements such as "get darker", "when the user is sleeping" and "turn off the bedroom light." These endeavors alleviate the burden on end users by liberating them from concerns pertaining to specific devices. Instead, they can focus more on defining their goals and values to support their lifestyle choices, as envisioned by Mennicken et al. for future smart homes in [55]. Consequently, they represent significant strides towards the overarching objective of accurately expressing end users' intentions. However, However, the interrelationships among these levels of requirement abstraction and their connections with TAP rules remain elusive. Despite the EUPont ontology [19] making strides in linking certain high-level requirements (device-independent functionalities) with low-level requirements (specifying only device types) [19, 20], there is still much to be clarified. On one hand, this indicates the potential to forge connections between different levels of abstracted requirements. On the other hand, it also underscores the feasibility of leveraging domain-specific knowledge to establish a connection from high-level needs to TAP rules.

In the year 2020, we conducted an online survey to gain insights into end users' preferences regarding the utilization of concrete device scheduling logics for expressing their needs [10]. Based on the findings from that study, this paper systematically investigates the expression of end users' intentions in the context of smart homes.To achieve this, we use a mixed-method approach that include a systematic mapping study (SMS) along with two common requirements elicitation techniques in requirements engineering (RE) [58]: questionnaires and interviews. Leveraging the insights garnered through this comprehensive investigation, and inspired by the architecture of smart homes, we have identified the needs of smart homes and their contexts, categorizing them into three interrelated levels of abstraction: high-level end users' intentions, medium-level intention realization plans, and low-level device scheduling logic. Based on these insights, we propose an innovative multi-level language known called *SH-RDL*. The aim of this language is to empower end users to articulate various intentions related to everyday activities in an easy and effective manner. At the same time, it can also serve as inputs to automatically transform user expressions into executable device scheduling logic such as TAP rules. The adoption of this language has immense potential not only for simplifying the articulation of user intentions in TAP but also for addressing the inherent limitations of TAP rules. This could potentially encourage a broader application of IoT end-user programming.

This paper further highlights the challenges inherent in the usage of the proposed language. The main challenge lies in automatically transforming higher-level intentions into executable device instructions, trying to bridge the gap between user-friendly expressions and practical implementation. We also point out that a possible solution is to utilize the contextual knowledge of requirements at various levels to achieve transitions between different levels of requirements. To validate the feasibility of the automatic transformation, we have conducted preliminary research. The results confirm its effectiveness in supporting high-level requirement programming. Moreover, an evaluation encompassing usability, understandability, and error-prevention is performed, serving to emphasize the notable benefits engendered by the utilization of this language. These results provide empirical evidence supporting the potential benefits of incorporating intentions within smart home automation systems.

The rest of this paper is organized as follows. Section 2 provides an discussion on the related work. In Section 3, the mixed-method approach for conducting our survey is detailed, followed by an analysis of the survey data and the presentation of obtained results. Section **??** presents the requirements description language *SH-RDL*. Furthermore, in Section **??**, challenges are presented, along with experiments carried out to demonstrate the feasibility for automatic requirements level transformations. An evaluation of the language itself is conducted in Section **??**. Finally, Section **??** concludes this paper, summarizes the key findings and suggests future directions for research in this domain.

2 RELATED WORK

This paper embarks on an exploration of TAP rules, with the intention of introducing a novel requirements description language designed to simplify inputs for end users in TAP. The research builds upon existing techniques for simplifying TAP and requirements languages employed in RE.

2.1 Simplifying Trigger-Action Programming

To simplify the TAP, extensive research has been conducted. Within this domain, the technical perspective has been profoundly explored. For example, some studies have dedicated themselves to harnessing the power of visual interfaces. Multiple visualization tools have surfaced, supporting users in expressing TAP rules through configurable wizards, wires, building blocks, and other intuitive mechanisms [6, 29, 31, 36]. Some researchers have explored programming by example or demonstration [8, 30]. Natural language is also utilized as a means of writing TAP rules. The natural-language descriptions are mapped into simple "if-then" rules through machine learning [50, 62]. However, despite the technical assistance provided by these studies, the problems they address are confined to the construction of TAP rules, and the level of requirements remains at the TAP rule level.

Lower level expressions, such as traces are also leveraged to generate TAP rules. For example, Zhang et al. introduce Trace2TAP, which automatically synthesizes TAP rules from time-stamped logs containing sensor readings and manual device actuations [79]. Their algorithm employs symbolic reasoning and SAT-solving techniques to derive TAP rules from the provided traces. Similarly, Liu et al. propose TAGen, which automates TAP rule generation by mining historical event traces [51]. TAGen enhances event traces with contextual information and identifies frequent event pairs to infer potential conditions. It then applies heuristic techniques to filter and rank candidate rules. While synthesizing TAP rules from traces significantly reduces the burden on users, there is still a degree of uncertainty in precisely capturing user intentions, and it cannot guarantee successful recognition every time.

There have also been efforts to integrate user intentions into the writing of TAP rules. Some have considered crowdsourcing, developing crowd-powered systems like InstructableCrowd [38], which allows users to program their devices through conversational interfaces. Users can articulate their needs verbally, and crowd workers help to establish the corresponding IF-THEN rules. There are also platforms that use recommendation systems, such as TAPrec [21] and HeyTAP [22], which utilize user intentions to recommend TAP rules. Users are able to express their personalized intentions and preferences through interaction with a conversational agent. Despite these advancements, crowdsourcing is still a manually intensive process, and recommendation systems often predict user intentions probabilistically rather than directly asking users.

Several studies have explored the use of high-level abstract requirements and their automatic transformation into TAP rules. For example, Ghiani et al. employed an "IF/WHEN trigger DO action" format to express device-independent intentions [35]. They mapped these high-level requirements directly to the underlying TAP implementation using a domain context model that encompasses information about sensors, devices, and their relationships. Zhang et al. described high-level needs using properties such as "air conditioning opening and window opening cannot occur together" and synthesized TAP rules to satisfy these properties [78]. Similarly, our previous work [11] utilized device states in the IoT system to represent high-level user requirements and developed an automated specification process from environment ontology to obtain TAP rules. These studies demonstrate the usefulness of abstract requirements representations in expressing users' personalized requirements, and the feasibility of transforming high-level requirements to low-level requirements. However, they mainly focus on limited requirements abstraction levels. In contrast, our paper aims to systematically uncover abstraction levels of requirement descriptions within smart homes which can be finally transformed into TAP rules. This endeavor will lay a solid foundation for the extensive application of TAP.

2.2 Requirements Languages in Requirements Engineering

Requirements Engineering (RE) is the process of discovering the purpose for the software, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation [58]. Requirements language can be divided into heavy modeling and lightweight description.

2.2.1 Heavy requirements modeling languages. There are various conventional requirements modeling languages, each representing different perspectives on requirements. One such example is KAOS [27], a goal-oriented modeling approach that captures the intentions of stakeholders through goals and represents the realization of goals via tasks. The process involves decomposing goals into sub-goals or tasks through the use of AND/OR relations until they can be effectively implemented by concrete tasks. Likewise, the i* framework [77] embraces the utilization of goals and sub-goals. It also endeavors to articulate the concept of "distributed intentionality". The goals and sub-goals are assigned to agents or software-under development through reasoning to derive software behaviors. In both models, goals serve as the medium for articulating requirements. Through refining goals to sub-goals, a spectrum of abstraction levels is achieved, including user intentions and low level software requirements.

The Problem Frames approach [41], presents a problem-centric approach that entails the classification, analysis, and structuring of software development problems. This approach involves identifying entities that will interact with the to-be built software, specifying requirements through expected phenomena that will happen on these entities, and subsequently deriving corresponding software behaviors according to entities' properties. Given its emphasis on the interaction between software and external entities, it is particularly well-suited for delineating the requirements of IoT systems [43]. It objectifies the level of requirements by contextualizing software requirements within the interactions between software and devices, while situating higher-level requirements within the interactions between devices and their external entities. It also points out that through domain knowledge, requirements at different levels can be progressively transformed into the most fundamental software requirements.

The aforementioned models have found utility in addressing IoT requirements. For instance, IoTReq combines KAOS and UML to elicit precise specifications for an IoT system [63]. It employs UML to model the domain, and then decomposes goals into sub-goals until the final goals of the IoT system can be achieved. Moreover, TrUStAPIS [33] also adopts a goal-driven approach. It defines an IoT requirement as a tuple comprising the contextual information, actor, action, and goal. The i* framework is also applied to the IoT domain. For example, Alqassem et al. [4] utilize goals to analyze privacy and security requirements. Our previous work [11] utilized the problem diagram of Problem Frames to express high-level user requirements and developed an automated specification process from environment ontology to obtain TAP rules.

To sum up, the above models are primarily designed for professional users rather than end users, making them challenging for non-professionals to grasp. Despite this, the research efforts underscore the significance of considering requirements across a spectrum of abstraction levels, spanning from user needs to software requirements. Indeed, it is feasible to transform high-level concepts into low-level abstractions though domain knowledge.

2.2.2 Lightweight requirements description languages. Several lightweight requirements description languages have been proposed in the literature to facilitate the description of requirements. Notably, Mavin et al. propose the Easy Approach to Requirements Syntax (EARS) method [53], which introduces a simple syntax: "While <optional precondition>, when <optional trigger>, the <system name> shall <system response>". EARS operates as a template language, incorporating six templates that can accommodate diverse requirements, including safety requirements. It is characterized by simplicity, flexibility, and ease of use, making it suitable for adoption across various domains [52].

The investigation of requirement boilerplates, a form of semi-formal representation, has gained significant attention from both academic research and industry practitioners [32]. Boilerplates consist of attributes and fixed syntax elements (FSEs) that contribute to their standardized structure. For instance, a common boilerplate configuration is "<system> shall <action>", with <system> and <action> representing attributes and <shall> serving as an FSE.

Alternative semi-formal representations have also been proposed. For instance, Verma et al. introduce a controlled syntax set encompassing both standard requirements and conditional requirements syntax within a tool named Requirements Analysis Tool (RAT) [75]. The commonly used standard requirements syntax for composing requirements follows the structure: <agent> <modal word> <action> <rest>. There's also a conditional requirements syntax, which is structured as if <condition> then <StandardRequirement>. This format allows for the specification of requirements that depend on certain conditions.

In conclusion, requirements templates or boilerplates offer constrained grammars for specifying detailed requirements, making them easily mastered for end users. Despite their proficiency in addressing low-level (operational) requirements, they have yet to fully extend to the higher levels of abstraction essential for capturing the breadth and depth of the comprehensive user needs spectrum. Hence, discerning and delineating distinct levels of requirements that can naturally encapsulate and express the diverse and nuanced needs beyond the operational scope is of paramount importance for propelling the advancement of smart home TAP and for the evolution of requirement languages themselves.

3 RESEARCH METHODS AND RESULTS

The mixed-methods study consists of three phases. The initial phase comprises an SMS (Systematic Mapping Study) to ascertain the various abstraction levels of requirements addressed in existing researches and their corresponding descriptions. Subsequently, an online survey is conducted to gauge the preference of end users with regards to different requirement levels in practical scenarios using a questionnaire. Finally, a requirements interview is conducted to delve into the expression patterns and key elements through which end users articulate their needs in real-life scenarios. Notably, for the online survey and offline interview, we have formulated a research ethics protocol that has been submitted to and approved by the Research Ethics Board of our university. Prior to participation, participants are duly informed about the study's objectives and are required to provide their informed consent by signing a consent form.

3.1 Systematic Mapping Study

3.1.1 Method. Following the established guidelines for conducting an SMS [61], we employed an automatic search strategy supplemented by manual searches through conference proceedings and targeted journals focusing on the IoT, RE, and HCI community. In formulating our search keywords, we focused on three core elements: domain, user, and topic. The domain was confined to the IoT, TAP, or smart homes. For user, we considered roles such as users, consumers, and end users. The topics encompassed aspects like requirements, needs, concerns, intentions, perspectives, preferences, development, and planning. Based on these elements, we constructed the following search keyword combinations: ("IoT" OR "internet of things" OR "IF-THEN rules" OR "Trigger Action programming" OR "Trigger-Action rules" OR "TAP" OR "smart home") AND (requirement? OR need? OR concern OR intention* OR perspective? OR preference? OR development? OR program) AND (person* OR user? OR end-user? OR people OR consumer? OR customer?).

We have established three inclusion criteria (IC) to determine the inclusion of primary studies in our analysis, and four exclusion criteria (EC) to ensure the relevance and quality of the selected studies.

IC_1: Primary studies necessitates the incorporation of users' needs within the context of smart homes. We adopt a broad perspective at this stage to encompass a wide range of relevant papers, ensuring the comprehensiveness of our study.

IC_2: Classic sources, including foundational books and seminal papers, to shape our understanding of the subject.

IC_3: Gray literature such as technical reports and preliminary studies in capturing user needs expressions.

EC_1: Studies that mainly address requirements verification, design or realization, as they fall outside our research scope.

EC_2: Duplicate studies on a given issue are not considered. Only the most comprehensive and complete paper pertaining to each unique issue is included.

EC_3: Studies that lack full access to their electronic versions, are excluded from our investigation.

EC_4: Papers written in other languages except English and Chinese are not considered due to our proficiency.

Data extraction is carried out by a four-member team including two master students, an associate professor, and a professor, all experts in IoT and RE. The initial phase of the process involved students selecting relevant studies by evaluating their titles and abstracts in accordance with criteria IC_1 to IC_3, excluding any that were not pertinent to the research focus. Subsequently, students conducted a thorough review of the selected papers and excluded those that did meet the criteria set by EC_1 through EC_4. Concurrently, professors conducted a re-examination of the excluded papers. To ensure a rigorous selection process, each paper was evaluated by a minimum of two authors, including at least one professor. In the event of disagreements, all authors would jointly assess the paper and reach a consensus through discussion. Finally, students extracted key information from the papers that were ultimately selected, focusing particularly on specific requirement expressions and types.

3.1.2 Results Analysis. After an initial automatic search, a comprehensive collection of 143,501 papers was obtained, with significant contributions from various databases: 36,094 from the IEEE Xplore Digital Library, 31,111 from the ACM Digital Library, 39,940 from the Springer Link Digital Library, and 36,358 from the combined resources of the Web of Science and Engineering Village. Following a meticulous review of the titles and abstracts, this extensive list was narrowed down to 151 papers. Ultimately, after a thorough reading of the full texts, a select group of 32 papers was chosen for further analysis.

Through these papers, our aim is to delve into the levels of abstraction of requirements. Given the existence of various abstractions, there is a need to establish a unified standard for analysis. The perspective of requirements from the Problem Frames approach suggests that different levels of abstraction are grounded in different contexts, a point corroborated by the research of Corno and colleagues[23]. To conduct a comprehensive analysis of diverse contexts, we have considered the architecture of the smart home system, as depicted in Fig. 1. The smart home system is designed to interact directly with devices, including sensors and actuators. Additionally, it engages in indirect interactions with the natural environment, humans, devices, and external entities such as the flooring. Based on this architecture, we have identified three related levels of requirements:

- High Level: The context includes the environment, humans, and external entities. The description focuses on anticipated changes in the environment or external entities, or the exchange of information with humans, independent of the devices. For instance, the temperature should be set at 25 degrees Celsius.
- Medium Level: The context should encompass devices or specific types of devices, the environment, humans, and external entities. The description is centered on the expected interactions between the devices and the environment, humans, and external entities, related to the type of device. For example, this could involve the status of devices or the signals they emit.
- Low Level: The context should be the software and specific devices with a particular focus on communication features. The description is concerned with the interactions between the software controller and the devices, which can include commands issued to the devices.

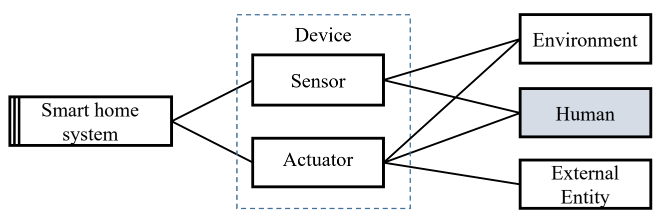


Fig. 1. Architecture of smart home systems

Based on these three abstract levels, the classification of the aforementioned papers was conducted. The results are presented in Table 1. Notably, from the "Requirement Expressions" column, we can see requirement descriptions in $S_1 - S_{10}$, which mainly consisting of specific triggers and actions designed for specific devices or services, are classified as low level. Even though some requirements, like S_5 and S_{10}, include device-independent triggers like a person being home, they are still deemed as low-level requirements due to the specific actions linked with them.

Among the requirements listed, $S_{11} - S_{14}$ exhibit a combination of both medium and high level abstraction. They both have device-independent expressions such as "get darker" and device type-independent expressions such as " turn lights off". The requirements specified in $S_{15} - S_{28}$, alongside the comfort factor outlined in S_9, are non-functional factors

Table 1. Selected studies in the SMS. "ID" is the unique paper identifier, "Ref." is the corresponding reference, "Con.D/S" denotes concrete devices and services, "T/B-ind." indicates technology and brand independence, and "D-ind." denotes device independence.

ID	Ref.	Requirements expression	Con.D/S	T/B-ind.	D-ind.	Abstraction level
S1	[42]	if-this-then-that style	✓	X	X	
S2	[68]	Amazon Alexa(Routines)	✓	X	X	
S3	[24]	AppsGate's programming language	✓	X	X	
S4	[65]	triggers and subsequent actions	✓	X	X	
S5	[60]	TAP rules	✓	X	✓	low
S6	[64]	smart home rules	✓	X	X	
S7	[69]	certain extensions to TAP rules	✓	X	X	
S8	[36]	Extended rules with random time trigger	✓	X	X	
S9	[47]	"if-then" rules or "routines" for their devices	✓	X	X	
S10	[29]	Extended event-condition-action rules	✓	X	✓	
S11	[19]	Abstract IF-THEN rules	X	✓	✓	
S12	[20]	High-level TAP rules	X	✓	✓	medium
S13	[22]	Abstract IF-THEN rules	X	✓	✓	and high
S14	[35]	Abstract event-condition-action rules	X	✓	✓	
S9	[47]	Comfort	X	X	✓	
S15	[2]	Security and safety	X	X	✓	
S16	[18]	Energy consuming	X	X	✓	
S17	[3]	Security	X	X	✓	
S18	[37]	Security, safety and privacy	X	X	✓	
S19	[25]	Safety (negative expressions)	X	X	✓	
S20	[72]	Security, safety and privacy	X	X	✓	
S21	[45]	Security, safety and privacy	X	X	✓	high
S22	[26]	Security, safety and privacy	X	X	✓	
S23	[7]	Privacy	X	X	✓	
S24	[70]	Privacy and trust	X	X	✓	
S25	[14]	Security, safety and privacy	X	X	✓	
S26	[13]	Security, safety and privacy	X	X	✓	
S27	[13]	Security, safety and privacy	X	X	✓	
S28	[16]	Privacy	X	X	✓	

that do not rely on any specific device. Given their nature, they are at a high level of abstraction. The non-functional factors refer to the characteristics of a software system that are not related to specific functionality or behavior, as defined by Chung et al. [17]. These factors manifest as constraints or requirements imposed upon the system. It is worth noting that some non-functional factors can be realized through the implementation of functional requirements. For instance, the notion of safety can be translated into a functional requirement, such as "When smoke is detected, send an alarm", which directly relates to system behavior. Our survey indicates that safety, security, and privacy are the primary concerns in the field of smart homes. When it comes to these non-functional factors, individuals often express their needs using different linguistic expressions that align with their personal habits. Interestingly, S_{19} reveals that users often use negative expressions like "I don't want to..." or "...should never occur" to articulate their safety requirements.

Based on our analysis, we have discovered that existing requirement descriptions have encompassed three levels of abstraction. Although most papers may only address one or two levels, they do not fully cover all three. The primary expression pattern for these requirement levels continues to be the IF-Then sentence structure, and numerous non-functional requirements have also emerged, such as comfort, privacy, safety, and energy conservation. We view these requirements as fundamental to human and consider them a manifestation of high-level needs. This indicates that while the current work may implicitly include the three levels of abstraction, there may be differences in the forms of expression, and the intrinsic connections between these levels of abstraction have not been revealed through an architectural perspective.

3.2 Online Survey

3.2.1 Method. The first step involves designing a questionnaire to investigate end users' preferences and familiarity with their devices in real-life situations. We have created four groups of questions. For more details, interested readers can find the questionnaire at http://re4cps.org/survey.

- The first group of questions focuses on gathering demographic information from the participants. It includes six specific inquiries about gender, age, marital status, educational background, relevance of their major to computer science or software engineering, and familiarity with smart home-related knowledge.
- The second group of questions evaluates the participants' proficiency in interacting with smart devices. This set of three questions examines their understanding of the functionalities of both traditional and cutting-edge devices, as well as their awareness of energy consumption.
- The third group of questions seeks to understand how participants usually express their functional requirements for smart homes. It includes five multiple-choice questions where users are asked to distinguish between device-dependent and device-independent expressions that convey the same meaning. Additionally, two questions inquire about their confidence in selecting the appropriate device(s) for specific scenarios. Lastly, three open-ended questions allow respondents to freely express their requirements based on their own judgment.
- The fourth group of questions explores user preferences regarding non-functional factors such as safety, security, privacy, and energy consumption. These factors are often overlooked in many situations. To understand these preferences in detail, we have created three questions that aim to capture users' natural expressions. These questions cover different aspects of end-user preferences and aim to identify patterns.

The online survey was conducted using an online questionnaire platform, *Sojump*, following a three-step process. Firstly, the questionnaire was distributed across various social media platforms and academic communities. Secondly, feedback from participants was collected. Finally, detailed analysis of the collected data was performed to derive meaningful insights. This process lasted for one month to allow sufficient time for participation. The participants came from diverse backgrounds including professionals such as teachers, doctors, students, workers, engineers, officials, farmers, and others. The survey takes participants an average of 5 minutes and they are compensated 1-3 RMB for their time. Throughout the duration of the survey, a total of 572 participants were initially invited to participate. Regrettably, 5 participants failed to complete the questionnaire, ultimately resulting in 567 valid feedback responses.

3.2.2 Statistics Analysis. The initial set of questions provides a comprehensive insight into the demographic data of the participants. Among the 567 respondents, 46.91% are female, while 53.09% are male. Their ages ranged from under 18 to over 71. Among them, the largest age group is 19-31 years old, accounting for 46% of the total, followed by the 31-40 age group, which accounts for 30%. In terms of educational background, 58.02% of participants hold bachelor's degrees, and 20.99% possess master's degrees. Regarding expertise in computer-related fields, 47.27% of participants have relevant majors. Among the respondents, only 1.41% self-identified as experts in smart homes, and most of them (51.32%) demonstrated some level of familiarity. The demographic data of the participants can be found at Appendix A. Based on the aforementioned statistics, the participant cohort is representative in terms of gender and age, and they possess a relatively high level of education and computer knowledge. Furthermore, it is important to highlight that our diverse participant pool includes individuals from various backgrounds.

From the second group of inquiries, we have derived the following conclusions. 54.32% participants lacked knowledge about additional functionalities of air conditioners beyond their basic cooling and heating capabilities. The 93.83% participants are not familiarity with alternative features of tower fans, a relatively new home appliance, indicating a lack of exposure or awareness of this particular device. Only 15.34% participants claimed to have a comprehensive understanding of the power consumption for most, if not all, of their home devices. Detailed data is available in Appendix A. These findings highlight a significant lack of familiarity among participants regarding their smart devices.

The third group is centered around comprehending the participants' preferences regarding functional requirements in smart homes. In the set of five single-choice questions, a minimum of 64.55% individuals preferred device-independent expressions. For each exact choice and proportion, please refer to Appendix A. This suggests that end-users have a predilection for device-independent terminologies. Furthermore, the subsequent two questions, pertaining to participants' certainty in utilizing specific devices under varying temperature conditions, yielded a majority response of "It depends" for both questions. This finding aligns with the conclusions drawn in [28], which posits that individuals aspire to exert control over their lifestyles rather than mere device management.

Participants were asked three open-ended questions to express their requirements in 3 different scenarios, namely the bedroom, kitchen, or study. The responses were categorized into three types. The first category includes requests for automated adjustment of environmental attributes, such as "automatically adjusting the brightness". The second type relates to device automation for direct control over specific devices. Participants often gave examples like "automatically opening the window when I come home". Lastly, participants showed significant interest in scenario-specific operations. For example, in the kitchen scenario, many respondents expressed desires for "automatic cooking". These requirements highlight abstract functionalities rather than being device-centric. Among the three scenarios, the study scenario mainly involves automated adjustment (42.50%), while device automation is most common in the bedroom scenario (40.74%). Scenario-specific operations are primarily observed in the kitchen scenario (59.08%). The exact distribution of these responses in different scenarios can be found at Appendix A.

The final group of questions aimed to uncover the non-functional aspects that participants consider important. Regarding the first two questions, an overwhelming majority of participants (97.71%) emphasized the importance of safety and security, while 95.06% advocated for the inclusion of privacy considerations. The third question prompted participants to articulate non-functional preferences beyond those related to safety, security, and privacy. Notably, 71.61% of participants highlighted the significance of environmental protection, while 91.83% underscored the need for a noise-free environment. Additionally, participants frequently mentioned other factors such as attractive device aesthetics, ease of use, and energy efficiency. These findings establish a comprehensive understanding of the non-functional factors.

In summary, our analysis of the online survey has led to several noteworthy findings. Firstly, we find that the majority of end users exhibit limited familiarity with their smart home devices, regardless of whether these devices belong to traditional or recently introduced categories. When formulating functional requirements, users typically avoid explicitly mentioning specific devices. Instead, they often use representations that depend on the device but are not tied to a particular brand, showing a moderate level of specificity in terms of the device type. When unsure about which device to use, users prefer to describe the desired environmental effects instead. These descriptions go beyond specific devices and operate at a higher level of abstraction. In addition to the traditional "IF-THEN" pattern, users frequently use the "should" pattern to express their opinions on whether certain actions should or should not happen. Lastly, we have observed the prevalence of two distinct levels of abstraction that commonly manifest in real-world scenarios:

- **Device dependent but brand-independent expressions** are abstract IF-THEN rules, focusing on device types rather than specific devices. For example, "IF the camera recognizes me, THEN turn on the lamp".
- **Device-independent expressions** refer to functions, environment effects, and preferences like energy-saving, which are unrelated to devices. Notably, these expressions are predominantly favored by end users.

3.2.3 Correlation Analysis. In this section, we explore the factors affecting users' preference for using device-independent expressions. At the beginning, we prepare the dataset from the online survey which encompasses a total of 567 samples. Specifically, we measure the frequency of device-independent expression usage by determining the proportion of respondents who select device-independent requirement expressions within the third group of inquiries. We manually categorize responses to this group of questions as either containing or lacking device-independent expressions. Subsequently, each answer provided in the first group of questions is transformed into an ordered integer, encompassing all demographic factors as variables.

To provide quantitative assess, we firstly employed Factor Analysis [44]. During the suitability test for factor analysis, the Kaiser-Meyer-Olkin (KMO) [44] test yielded a result of 0.53. This is below the generally accepted minimum of 0.6 for meaningful results, indicating insufficient association between variables. This negates the need for further single or multiple factor analyses. In addition, we further confirm this weak association through the value of correlation coefficients. We calculate the Spearman correlation coefficient [71] between the frequency of device-independent expression usage and all variables (details are depicted in Appendix A). It turns out that the correlation of all variables are close to zero, varying from -0.02 to 0.07, indicating a lack of substantial correlation between the frequency of device-independent requirement expressions and other variables.

To conclude, from the perspectives of both the KMO test and the correlation coefficients, collected demographic factors do not have a significant impact on users' tendency to use device-independent expressions. We can make a conclusion that **people prefer to use device-independent expression regardless their education background, age, knowledge of smart homes, major, gender and marriage**. Furthermore, we found in specific populations, there exist associations and differences in the expression of end user needs using stratified analysis [49]. We start by dividing the data set according to different combinations of factors as conditions. Dozens of potentially meaningful combinations, controlling single or multiple factors and their ranges, are tried to study the association between other factors and the preference of users' expression. We calculate the Spearman correlation coefficient as the main indicator.

Notably, we identify two distinct trends among users over the age of 40 with computer-related degrees, all of which display a moderate correlation, as indicated by absolute coefficient values ranging from 0.4 to 0.6 across 41 samples (details are in Appendix A). The first trend, signaled by a coefficient of -0.40, suggests that users with less smart home knowledge are inclined towards using device-independent expressions, likely due to their lower requirement for specific knowledge compared to device-dependent ones. The second trend, underscored by a coefficient of 0.45, indicates a preference among more educated users towards using device-independent expressions. This could be attributed to their capacity to effectively communicate their needs or a disinclination towards frequent device upgrades due to rapid technological advancements. Besides, for users under 40, we see a decrease in the strength of previous trends with younger ages, suggesting they are more open to new smart home technologies. Those without a computer-related major also have similar trends, but to a lesser extent. The aforementioned observation does not impact our previously drawn conclusion that people generally prefer device-independent expressions, and instead further validates the rationality of the data collected indirectly.

3.3 Requirements interview

3.3.1 Method. Our objective is to explore the practical aspect of how end users articulate their requirements, and analyze the fundamental elements present in their expressions. To this end, we design three open-ended questions for the participants, each relating to a distinct scenario: the living room, kitchen, and study room. The questions are: *"To enhance the convenience and comfort of your daily living and studying experiences, what functions or automation would you desire from smart devices? These could be devices you currently own or ones you hope to own in the future."* To facilitate communication, all our interviews are conducted in Chinese.

During our interview phase, we successfully recruited 99 participants. A portion of these participants were sourced from the social networks of our two student author, while the rest were brought in through recruitment efforts staged at the entrance of the university cafeteria. Although the majority of our interviews were conducted at the university cafeteria entrance, we also arranged a number of sessions at the participants' homes or student dormitories for their convenience. On average, participants dedicate approximately 10 minutes to complete the interview, and receive a 3 RMB gift as thanks. This interview lasted for a week.

During the interview process, two student authors take on different roles: one as the host, responsible for asking questions; the other as the recorder, responsible for collecting requirements. Prior to proceeding, participants are requested to provide consent for their answers to be recorded using audio devices. The host presents the participant with three questions, prompting them to contemplate their needs. After taking a brief moment to contemplate their requirements, the participant, once ready, allows the recorder to proceed with documenting their responses. Finally, these audio files are converted into textual data using the iFLYTEK speech recognition platform [1] for further analysis. Besides gathering responses, the recorder also collect additional demographic information such as gender, age, educational background, and familiarity with smart home devices. To better capture the expressions of the end users, we intentionally provide minimal guidance to participants to elicit their genuine expressions without any influence. However, the host may offer gentle reminders to consider non-functional requirements if they are unintentionally overlooked by participants.

3.3.2 Interview Results. Our sample size of 99 individuals comprises 57 males and 42 females. The age range spans from 18 to 71, with a predominant concentration (78 interviewees) falling within the youthful bracket of 18 to 30 years. Examining the educational background, 46 interviewees possess a bachelor's degree, while 37 hold a master's degree, attesting to the high academic qualifications of our participants. As to the participants' smart home knowledge, a substantial majority of 86 interviewees modestly ranked themselves a relatively limited understanding of smart homes. Only one interviewee claimed full knowledge, highlighting the prevailing gap in expertise among the respondents. These statistical observations (details in Appendix A) underscore that our participant pool exemplifies representative gender ratios while exhibiting youthful tendencies and a high education level.

In total, we amass 182 minutes of recorded data, encompassing 74 minutes devoted to the living room, 42 minutes focused on the kitchen, and 66 minutes centered around the study area. These data include 324 requirements for the living room, 211 for the kitchen, and 255 for the study area. Subsequently, a filtering process is conducted to eliminate any requirements deemed irrelevant to the software domain such as "my device must have an attractive appearance". Consequently, we arrive at 230, 149 and 139 requirements respectively. It is to be noticed that only 35% of the interviewees mentioned the non-functional requirements without reminding. This suggests that non-functional requirements are often overlooked by end users. The identified requirements encompass both functional and

[1] https://global.xfyun.cn/

non-functional factors, warranting a categorization scheme featuring two distinct abstractions: brand-independent representations and device-independent representations, akin to the structure in the online survey.

We conducted a statistical analysis of these expressions and found that from a syntactical perspective, the common patterns are 'if-then', 'when', and 'should'. This is consistent with our findings from the online survey. Further, we conduct a thematic analysis [12] of the core elements utilized to express users' needs in device-independent formats. During the analysis process, two researchers were assigned different tasks: one transcribed the user expressions from audio to text, and the other translated the text into English. After completing their respective tasks, each researcher reviewed the other's work for accuracy. Next, both researchers independently attempted to extract codes from the entire dataset, identifying elements within the sentences, sentence patterns for expressing needs, and so on. They then combined closely related content into themes. Finally, the researchers presented their individual analysis results to each other, discussed the potential value of these codes and themes for understanding smart home end users' preferences in expressing their needs, and integrated their findings. For processing, each requirement sentence is deconstructed into sub-clauses and translated into English. A distinctive feature of this procedure is the abstraction of certain specific expressions during the coding for cluster analysis. Specific devices are universally translated into abstract devices, references to individuals are converted into generalized human behaviors, smart home objects such as "floor" are morphed into external entities, precise non-functional requirements like "safety" are substituted by broader non-functional requirements, while temperature and humidity are encoded as environmental information. As a result, we created the thematic map shown in Fig. 2. Ultimately, this exploration reveals five unique elements that are commonly encountered:

- "The floor is clean" represents an *effect* coveted by users, signifying their desired outcome.
- "Automatic cooking" denotes a *function* that the smart home system must furnish.
- "Maintain the temperature around 26^oC" embodies a desired *value* pertaining to an *environment attribute* that necessitates regulation.
- "When I arrive home, dinner is ready" illustrates a coherent sequence of events wherein "I arrive home" encapsulates a *human behavior* while "dinner is ready" epitomizes the intended effect.
- Natural language expressions for non-functional requirements such as "I desire a comfortable home" or "I prefer energy-saving".

Moreover, with regard to the elemental components comprising device-dependent but brand-specific expressions, we identify four typical manifestations:

- The phrase "Turn on the air conditioner if the temperature exceeds 26^oC" embodies an *action specific to the device type* that the system performs, paired with an environmental attribute trigger.
- "My surveillance camera should always be on" emphasizes the requisite *device state* denoted by "on".
- "Activate the air conditioner in the living room 20 minutes before the camera recognizes me" integrates location information, such as "in the living room", along with a temporal element denoted by "20 minutes before".
- Statements like "don't open the window when no one is in my house" and "the heater and the air conditioner should not be turned on together." reveal the usage of negative expressions in requirements expression.

It is to be noticed that we particularly examined whether there were any differences in the expressions between the scenarios imagined by the participants in-situ (e.g., at home or student dormitories) and those imagined during interviews in the cafeteria. The results showed that their expressions were similar across both contexts.

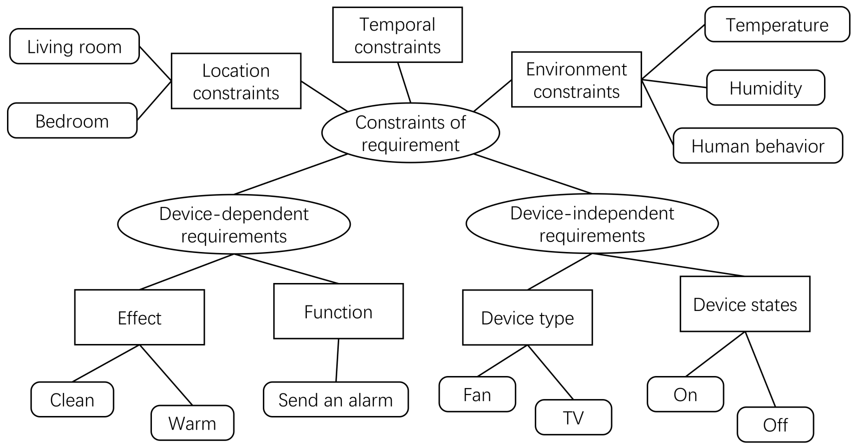


Fig. 2. Thematic map of the requirements of end users

3.4 Threat to validity

There are three potential validity threats. The first is the lack of diverse participant sampling. Our online survey was mainly filled out by highly educated individuals, and younger people mostly made up the requirements interviews. To offset this potential bias, we suggest gathering more data from less educated individuals for the online survey and from older individuals for the interviews. This would create a more diverse participant pool and likely lead to an increased proportion of device-independent requirements and a decreased proportion of device-dependent requirements. However, it's important to stress that these adjustments would not change our study's overall focus. We are confident that our research aims to illuminate how end users articulate their needs in the context of smart homes. While participant demographics may affect the volume of need expressions, they do not compromise the validity of our results.

The second potential threat to the validity of our study arises from the geographical and demographic limitation of conducting both the online and offline surveys exclusively within China, involving only Chinese nationals. Despite this constraint, it's important to note that our research aligns with prior investigations on end-user programming in real-world settings [23, 54, 76], as well as exploratory studies pertaining to future smart home programming [9, 57]. These complementary studies provide additional support and contextual grounding for our findings.

The third potential threat is rooted in the lack of evaluation for the questionnaire used in the online survey. Feedback indicates that the single-choice questions about device-dependent and device-independent requirements may have subtly guided participants' responses to the three open-ended questions. Specifically, this potential bias suggests that users might be influenced to use expressions similar to those in the multiple-choice options, possibly leading to a preference for intention-based expressions. To counter this effect, we conducted additional requirements interviews. These revealed that, while this influence did affect the balance of device-independent and device-dependent requirements, it did not significantly alter the overall expressions provided by end users.

REFERENCES

[1] Shohin Aheleroff, Xun Xu, Yuqian Lu, et al. 2020. IoT-enabled smart appliances under industry 4.0: A case study. *Adv. Eng. Informatics* 43 (2020), 101043.

[2] Ahlam Alami, Laila Benhlima, and Slimane Bah. 2018. A Study of Security Requirements in Wireless Sensor Networks for Smart Home Healthcare Systems. In *SCA*. ACM, 55:1–55:8.

[3] Waqar Ali, Ghulam Dustgeer, Muhammad Awais, and Munam Ali Shah. 2017. IoT based smart home: Security challenges, security requirements and solutions. In *ICAC*. IEEE, 1–6.

[4] Israa Alqassem. 2014. Privacy and security requirements framework for the internet of things (IoT). In *Companion Proceedings of the 36th ICSE*. 739–741.

[5] Inc Amazon.com. 2014. Amazon Alexa. https://developer.amazon.com/alexa.

[6] Carmelo Ardito, Paolo Buono, Giuseppe Desolda, and Maristella Matera. 2018. From smart objects to smart experiences: An end-user development approach. *Int. J. Hum. Comput. Stud.* 114 (2018), 51–68.

[7] Natã Miccael Barbosa, Joon S. Park, Yaxing Yao, and Yang Wang. 2019. "What if?" Predicting Individual Users' Smart Home Privacy Preferences and Their Changes. *Proc. Priv. Enhancing Technol.* 2019, 4 (2019), 211–231.

[8] Zouhir Bellal, Nadia Elouali, Sidi Mohamed Benslimane, and Cengiz Acartürk. 2021. Integrating Mobile Multimodal Interactions based on Programming By Demonstration8203. *Int. J. Hum. Comput. Interact.* 37, 5 (2021), 418–433.

[9] Andrea Bellucci, Andrea Vianello, Yves Florack, and Giulio Jacucci. 2016. Supporting the Serendipitous Use of Domestic Technologies. *IEEE Pervasive Comput.* 15, 2 (2016), 16–25.

[10] Han Bian, Xiaohong Chen, Zhi Jin, and Lin Liu. 2021. Smart3E: Enabling End Users to Express Their Needs for Smart Homes. In *29th IEEE International Requirements Engineering Conference*. IEEE, 422–423. https://doi.org/10.1109/RE51729.2021.00051

[11] Han Bian, Xiaohong Chen, Zhi Jin, and Min Zhang. 2021. Approach to Generating TAP Rules in IoT Systems Based on Environment Modeling. *Int. J. Softw. Informatics* 11, 3 (2021), 263–286.

[12] Virginia Braun and Victoria Clarke. 2012. *Thematic analysis.* American Psychological Association.

[13] George Chalhoub, Ivan Flechais, Norbert Nthala, Ruba Abu-Salma, et al. 2020. Factoring User Experience into the Security and Privacy Design of Smart Home Devices: A Case Study. In *CHI Extended Abstracts*. ACM, 1–9.

[14] George Chalhoub, Martin J. Kraemer, Norbert Nthala, and Ivan Flechais. 2021. "It did not give me an option to decline": A Longitudinal Analysis of the User Experience of Security and Privacy in Smart Home Products. In *CHI*. ACM, 555:1–555:16.

[15] Liangyu Chen, Chen Wang, Cheng Chen, Caidie Huang, Xiaohong Chen, and Min Zhang. 2024. TapChecker: A Lightweight SMT-Based Conflict Analysis for Trigger-Action Programming. *IEEE Internet Things J.* 11, 12 (2024), 21411–21426.

[16] Chola Chhetri and Vivian Genaro Motti. 2019. Eliciting Privacy Concerns for Smart Home Devices from a User Centered Perspective. In *iConference*, Vol. 11420. Springer, 91–101.

[17] Lawrence Chung, Brian A Nixon, Eric Yu, and John Mylopoulos. 2012. *Non-functional requirements in software engineering*. Vol. 5. Springer Science & Business Media.

[18] Mario Collotta and Giovanni Pau. 2017. An Innovative Approach for Forecasting of Energy Requirements to Improve a Smart Home Management System Based on BLE. *IEEE Trans. Green Commun. Netw.* 1, 1 (2017), 112–120.

[19] Fulvio Corno, Luigi De Russis, and Alberto Monge Roffarello. 2017. A Semantic Web Approach to Simplifying Trigger-Action Programming in the IoT. *Computer* 50, 11 (2017), 18–24.

[20] Fulvio Corno, Luigi De Russis, and Alberto Monge Roffarello. 2019. A high-level semantic approach to End-User Development in the Internet of Things. *Int. J. Hum. Comput. Stud.* 125 (2019), 41–54.

[21] Fulvio Corno, Luigi De Russis, and Alberto Monge Roffarello. 2020. TAPrec: supporting the composition of trigger-action rules through dynamic recommendations. In *IUI*. ACM, 579–588.

[22] Fulvio Corno, Luigi De Russis, and Alberto Monge Roffarello. 2021. From Users' Intentions to IF-THEN Rules in the Internet of Things. *ACM Trans. Inf. Syst.* 39, 4 (2021), 53:1–53:33.

[23] Fulvio Corno, Luigi De Russis, and Alberto Monge Roffarello. 2022. How do end-users program the Internet of Things? *Behav. Inf. Technol.* 41, 9 (2022), 1865–1887.

[24] Joëlle Coutaz and James L. Crowley. 2016. A First-Person Experience with End-User Development for Smart Homes. *IEEE Pervasive Comput.* 15, 2 (2016), 26–39.

[25] Maheswaree Kissoon Curumsing, Niroshinie Fernando, Mohamed Abdelrazek, Rajesh Vasa, et al. 2019. Emotion-oriented requirements engineering: A case study in developing a smart home system for the elderly. *J. Syst. Softw.* 147 (2019), 215–229.

[26] Nana Kesewaa Dankwa. 2020. Investigating Smart Home Needs for Elderly Women Who Live Alone. An Interview Study. In *HCI (49) (Communications in Computer and Information Science, Vol. 1294)*. Springer, 32–38.

[27] Anne Dardenne, Axel Van Lamsweerde, and Stephen Fickas. 1993. Goal-directed requirements acquisition. *Science of computer programming* 20, 1-2 (1993), 3–50.

[28] Scott Davidoff, Min Kyung Lee, Charles Yiu, John Zimmerman, et al. 2006. Principles of Smart Home Control. In *UbiComp*, Vol. 4206. Springer, 19–34.

[29] Giuseppe Desolda, Carmelo Ardito, and Maristella Matera. 2017. Empowering End Users to Customize their Smart Environments: Model, Composition Paradigms, and Domain-Specific Tools. *ACM Trans. Comput. Hum. Interact.* 24, 2 (2017), 12:1–12:52.

[30] Anind K. Dey, Raffay Hamid, Chris Beckmann, Ian Li, and Daniel Hsu. 2004. a CAPpella: programming by demonstration of context-aware applications. In *Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI*, Elizabeth Dykstra-Erickson and Manfred Tscheligi (Eds.). ACM, 33–40.

[31] Anind K. Dey, Timothy Sohn, Sara Streng, and Justin Kodama. 2006. iCAP: Interactive Prototyping of Context-Aware Applications. In *Pervasive*, Vol. 3968. Springer, 254–271.

[32] Stefan Farfeleder, Thomas Moser, Andreas Krall, Tor Stålhane, Herbert Zojer, and Christian Panis. 2011. DODT: Increasing requirements formalism using domain ontologies for improved embedded systems development. In *14th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems*. IEEE, 271–274.

[33] Davide Ferraris and Carmen Fernandez-Gago. 2020. TrUStAPIS: a trust requirements elicitation method for IoT. *International Journal of Information Security* 19, 1 (2020), 111–127.

[34] Mathias Funk, Lin-Lin Chen, Shao-Wen Yang, and Yen-Kuang Chen. 2018. Addressing the need to capture scenarios, intentions and preferences: Interactive intentional programming in the smart home. *International Journal of Design* 12, 1 (2018), 53–66.

[35] Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. 2017. Personalization of Context-Dependent Applications Through Trigger-Action Rules. *ACM Trans. Comput. Hum. Interact.* 24, 2 (2017), 14:1–14:33.

[36] Mateus Carvalho Gonçalves, Otávio Lara, Raphael Winckler de Bettio, and André Pimenta Freire. 2021. End-user development of smart home rules using block-based programming: a comparative usability evaluation with programmers and non-programmers. *Behav. Inf. Technol.* 40, 10 (2021), 974–996.

[37] Simon Himmel and Martina Ziefle. 2016. Smart Home Medical Technologies: Users' Requirements for Conditional Acceptance. *i-com* 15, 1 (2016), 39–50.

[38] Ting-Hao K. Huang, Amos Azaria, Oscar J. Romero, and Jeffrey P. Bigham. 2019. InstructableCrowd: Creating IF-THEN Rules for Smartphones via Conversations with the Crowd. *Hum. Comput.* 6 (2019), 113–146.

[39] Apple Inc. 2014. Apple HomeKit. https://developer.apple.com/homekit/.

[40] SmartThings Inc. 2012. Samsung SmartThings. https://www.smartthings.com/.

[41] Michael A. Jackson. 2000. *Problem Frames - Analysing and Structuring Software Development Problems.* Pearson Education.

[42] Timo Jakobi, Gunnar Stevens, Nico Castelli, Corinna Ogonowski, et al. 2018. Evolving Needs in IoT Control and Accountability: A Longitudinal Study on Smart Home Intelligibility. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4 (2018), 171:1–171:28.

[43] Zhi Jin. 2018. *Environment Modeling-Based Requirements Engineering for Software Intensive Systems.* Morgan Kaufmann.

[44] Henry F Kaiser. 1970. A second generation little jiffy. (1970).

[45] Sunwoo Kim and Jeonghyuk Yoon. 2016. An Exploratory Study on Consumer's Needs on Smart Home in Korea. In *HCI (18)*, Vol. 9748. Springer, 337–345.

[46] Amy J. Ko, Robin Abraham, Laura Beckwith, Alan F. Blackwell, Margaret M. Burnett, Martin Erwig, Christopher Scaffidi, Joseph Lawrance, Henry Lieberman, Brad A. Myers, Mary Beth Rosson, Gregg Rothermel, Mary Shaw, and Susan Wiedenbeck. 2011. The state of the art in end-user software engineering. *ACM Comput. Surv.* 43, 3 (2011), 21:1–21:44.

[47] Vinay Koshy, Joon Sung Park, Ti-Chung Cheng, and Karrie Karahalios. 2021. "We Just Use What They Give Us": Understanding Passenger User Perspectives in Smart Homes. In *CHI*. ACM, 41:1–41:14.

[48] Chieh-Jan Mike Liang, Börje F Karlsson, Nicholas D Lane, Feng Zhao, Junbei Zhang, Zheyi Pan, Zhao Li, and Yong Yu. 2015. SIFT: building an internet of safe things. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks.* 298–309.

[49] Shuhuang Lin, Yukun Ma, and Zunnan Huang. 2020. Advanced stratification analyses in molecular association meta-analysis: methodology and application. *BMC Medical Research Methodology* 20, 1 (2020), 1–14.

[50] Chang Liu, Xinyun Chen, Eui Chul Richard Shin, Mingcheng Chen, and Dawn Xiaodong Song. 2016. Latent Attention For If-Then Program Synthesis. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 4574–4582.

[51] Liwei Liu, Wei Chen, Lu Liu, Kangkang Zhang, Jun Wei, and Yan Yang. 2021. TAGen: Generating Trigger-Action Rules for Smart Homes by Mining Event Traces. In *Service-Oriented Computing - 19th International Conference, ICSOC 2021, Virtual Event, November 22-25, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 13121)*, Hakim Hacid, Odej Kao, Massimo Mecella, Naouel Moha, and Hye-young Paik (Eds.). Springer, 652–662.

[52] Alistair Mavin Mav and Philip Wilkinson. 2019. Ten years of EARS. *IEEE Software* 36, 5 (2019), 10–14.

[53] Alistair Mavin, Philip Wilkinson, Adrian Harwood, and Mark Novak. 2009. Easy Approach to Requirements Syntax (EARS). In *RE*. IEEE Computer Society, 317–322.

[54] Sarah Mennicken and Elaine M. Huang. 2012. Hacking the Natural Habitat: An In-the-Wild Study of Smart Homes, Their Development, and the People Who Live in Them. In *Pervasive*, Vol. 7319. Springer, 143–160.

[55] Sarah Mennicken, Jo Vermeulen, and Elaine M. Huang. 2014. From today's augmented houses to tomorrow's smart homes: new directions for home automation research. In *The 2014 ACM Conference on Ubiquitous Computing, UbiComp.* ACM, 105–115.

[56] Xianghang Mi, Feng Qian, Ying Zhang, and XiaoFeng Wang. 2017. An empirical characterization of IFTTT: ecosystem, usage, and performance. In *Internet Measurement Conference.* ACM, 398–404.

[57] Mahda Noura, Sebastian Heil, and Martin Gaedke. 2020. Natural language goal understanding for smart home environments. In *IoT '20: 10th International Conference on the Internet of Things, Malmö, Sweden, October*, Paul Davidsson and Marc Langheinrich (Eds.). ACM, 1:1–1:8.

[58] Bashar Nuseibeh and Steve M. Easterbrook. 2000. Requirements engineering: a roadmap. In *22nd International Conference on on Software Engineering, Future of Software Engineering Track, ICSE*, Anthony Finkelstein (Ed.). ACM, 35–46.

[59] Steven Ovadia. 2014. Automate the Internet With "If This Then That" (IFTTT). *Behavioral & Social Sciences Librarian* 33, 4 (2014), 208–211.

[60] Mitali Palekar, Earlence Fernandes, and Franziska Roesner. 2019. Analysis of the Susceptibility of Smart Home Programming Interfaces to End User Error. In *IEEE Symposium on Security and Privacy Workshops*. 138–143.

[61] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Inf. Softw. Technol.* 64 (2015), 1–18.

[62] Chris Quirk, Raymond J. Mooney, and Michel Galley. 2015. Language to Code: Learning Semantic Parsers for If-This-Then-That Recipes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL*. The Association for Computer Linguistics, 878–888.

[63] Gianna Reggio. 2018. A UML-based proposal for IoT system requirements specification. In *MiSE@ICSE*. ACM, 9–16.

[64] Michaela Reisinger, Johann Schrammel, and Peter Fröhlich. 2017. Visual end-user programming in smart homes: Complexity and performance. In *VL/HCC*. IEEE Computer Society, 331–332.

[65] Michaela R. Reisinger, Johann Schrammel, Stefan Suette, and Peter Fröhlich. 2020. Home Smart Home: Approachable Interfaces for Intelligibility, Modification, and End-User Programming. *IxD&A* 45 (2020), 226–245.

[66] Luigi De Russis and Fulvio Corno. 2015. HomeRules: A Tangible End-User Programming Interface for Smart Homes. In *CHI Extended Abstracts*. ACM, 2109–2114.

[67] Luigi De Russis and Alberto Monge Roffarello. 2018. A Debugging Approach for Trigger-Action Programming. In *CHI Extended Abstracts*. ACM.

[68] Ana-Maria Salai, Glenda Cook, and Lars Erik Holmquist. 2021. IntraVox: A Personalized Human Voice to Support Users with Complex Needs in Smart Homes. In *INTERACT (1)*, Vol. 12932. Springer, 223–244.

[69] Antti Salovaara, Andrea Bellucci, Andrea Vianello, and Giulio Jacucci. 2021. Programmable Smart Home Toolkits Should Better Address Households' Social Needs. In *CHI*. ACM, 718:1–718:14.

[70] Eva-Maria Schomakers, Hannah Biermann, and Martina Ziefle. 2021. Users' Preferences for Smart Home Automation - Investigating Aspects of Privacy and Trust. *Telematics Informatics* 64 (2021), 101689.

[71] Charles Spearman. 1961. The proof and measurement of association between two things. (1961).

[72] Kaiwen Sun, Yixin Zou, Jenny Radesky, Christopher Brooks, et al. 2021. Child Safety in the Smart Home: Parents' Perceptions, Needs, and Mitigation Strategies. *Proc. ACM Hum. Comput. Interact.* 5, CSCW2 (2021), 1–41.

[73] Linden Tibbets, Alexander Tibbets, Jesse Tane, and Scott Tong. 2010. IFTTT. https://ifttt.com/.

[74] Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. 2014. Practical trigger-action programming in the smart home. In *CHI*. ACM, 803–812.

[75] Kunal Verma and Alex Kass. 2008. Requirements analysis tool: A tool for automatically analyzing software requirements documents. In *International semantic web conference*. Springer, 751–763.

[76] Jong-bum Woo and Youn-Kyung Lim. 2015. User experience in do-it-yourself-style smart homes. In *UbiComp*. ACM, 779–790.

[77] Eric Yu. 2011. Modeling Strategic Relationships for Process Reengineering. In *Social Modeling for Requirements Engineering*. MIT Press, 11–152.

[78] Lefan Zhang, Weijia He, Jesse Martinez, Noah Brackenbury, et al. 2019. AutoTap: synthesizing and repairing trigger-action programs using LTL properties. In *ICSE*. IEEE / ACM, 281–291.

[79] Lefan Zhang, Weijia He, Olivia Morkved, Valerie Zhao, Michael L. Littman, Shan Lu, and Blase Ur. 2020. Trace2TAP: Synthesizing Trigger-Action Programs from Traces of Behavior. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 3 (2020), 104:1–104:26.

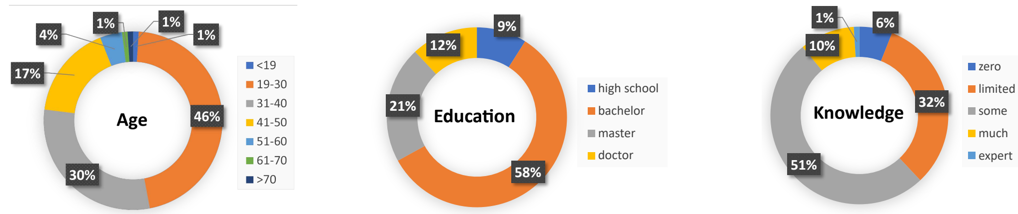
A DETAILS OF THE ONLINE SURVEY AND REQUIREMENTS INTERVIEW

In this appendix, we list parts of the results in the online survey and requirements interview.

A.1 Online survey

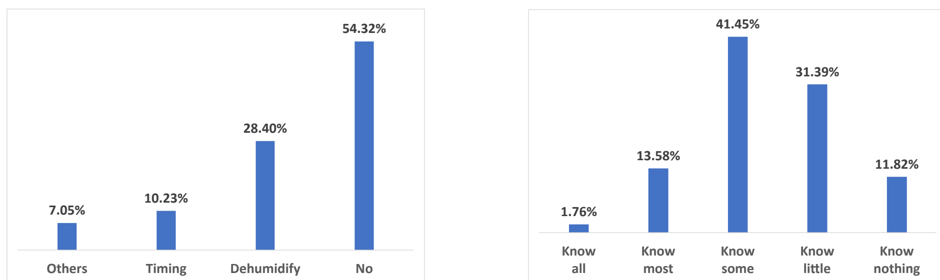
The aim of initial group of questions is to obtain a profound understanding of the participants' demographic data emerged. The exact data about age, educational background, and knowledge about smart homes are respectively listed in Fig. 3(a), Fig. 3(b), and Fig. 3(c).

In the second group of inquiries, the initial question sought to gauge participants' familiarity with common household devices. Specifically, respondents were asked whether they were aware of any functionalities beyond the standard cooling and heating capabilities of air conditioners. This statistical breakdown can be observed in Fig. 4(a). The subsequent query pertained to participants' acquaintance with alternative features of tower fans, a relatively novel home appliance. This question aimed to ascertain respondents' familiarity with emerging smart devices. The third question broached the subject of participants' awareness regarding the energy usage of their home devices. Fig. 4(b) presents the findings.



(a) age (b) education (c) knowledge

Fig. 3. Online survey: demographics analysis result



(a) knowledge about the smart devices (b) knowledge about energy consumption

Fig. 4. Online survey: participants' knowledge about smart devices and energy consumption

In the third group of inquiries, Table 2 showcases the responses to the initial five single-choice questions, elucidating that participants generally exhibit a preference for device-independent expressions. Fig. 5 illustrates the distribution of responses across different scenarios, i.e., bedroom, kitchen, and study in the three open-ended questions. It provides an informative visualization of the prevalence of each requirement type.

Table 2. Online survey: questions and answers of the six single-choice questions

ID	Choice A (device related expressions)	Choice A Proportion	Choice B (device-independent expressions)	Choice B Proportion
Q1	If the temperature is too high (low), then turn on the cold (hot) air conditioner.	20.63%	Maintain the temperature in the room at a certain value.	79.37%
Q2	If it is not raining, open the window.	35.45%	Ventilate the room when it is not raining.	64.55%
Q3	Start the sweeper every day.	31.92%	Keep the floor neat and tidy.	68.08%
Q4	The monitor should always be on.	9.56%	I want to monitor my home at any time.	90.44%
Q5	If it is raining, close the window.	29.15%	If it is raining, the window should be closed.	70.85%

In the correlation analysis, Fig. 6 depicts the heat map of the correlation between the frequency of device-independent expression usage, i.e., dev-indep, and all demographics variables. Although there is a strong correlation between certain variables, such as age and marriage, there is no significant correlation between these variables and the frequency of device-independent expressions.

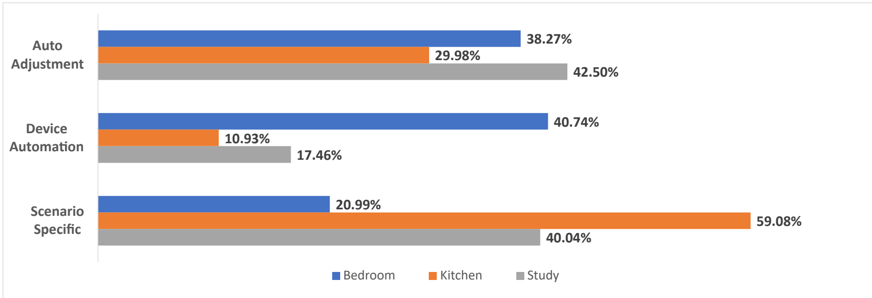


Fig. 5. Online survey: distribution of response in different scenarios

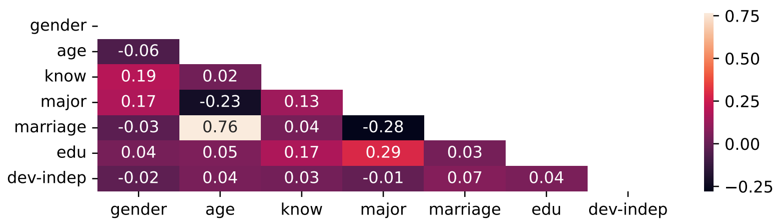
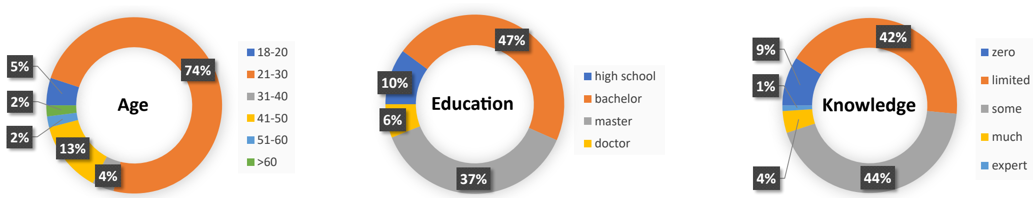


Fig. 6. Variables correlating with the frequency of using device-independent expression

A.2 Requirements interview

The demographics information is about age, educational background, and knowledge about smart homes. The exact data are respectively listed in Fig. 7(a), Fig. 7(b), and Fig. 7(c).



(a) age (b) education (c) knowledge

Fig. 7. Requirements interview: demographics analysis result

B DETAILS OF EXPERIMENT MATERIALS

Detailed supplementary materials including slides and videos used in introduction phase of the experiment is available on our website: http://re4cps.org/examples/sh-rdl. The tasks in the experiment, the demographic questionnaire, and the reference information mentioned in evaluation section are shown in the following subsections.

B.1 Tasks in the experiment

Task 1: requirements understanding

Please read the following requirements described using SH-RDL and try to understand their underlying intentions (i.e.what effect or function the user wants to achieve with these requirements).

e.g. 1: The intention of "surveillance_camera_on should hold" is "safety".

e.g. 2: The intention of "preferred room_temperature is 26 in living_room" is "Keep the temperature right".

- If time.6am, then make a cup of coffee
- Floor_clean should hold in living_room
- Preferred humidity is 26 in living_room
- CO_concentration should not be above 35mg/m^3
- If human.sleep, then curtain.close
- If stranger.outside then then remind homeowner.
- Projector.on and curtain.open should not occur together

Please read the following TAP rules and try to understand their underlying intentions (i.e., what effect or function the user wants to achieve with these requirements). ATTENTION: There may be cases where multiple TAP rules satisfying one intention.

e.g. The intention of "if human_sensor.reading = sleeping then close Philips Hue White and Color Ambiance" is "Feel comfortable during sleep".

- If temperature_sensor.reading < 24°C then turn_heat Daikin Inverter AC1000.
- If temperature_sensor.reading > 24°C and temperature_sensor.reading < 30 then turn_off Daikin Inverter AC1000.
- If temperature_sensor.reading > 30°C then turn_cool Daikin Inverter AC1000.
- If SmartMi window.open and turn_on Daikin Inverter AC1000, then close SmartMi window.
- If Daikin Inverter AC1000.on and open SmartMi window , then turn_off Daikin Inverter AC1000.
- If human_sensor.reading = no_human then turn_off Daikin Inverter AC1000.
- If human_sensor.reading = no_human then close Philips Hue White and Color Ambiance.
- If human_sensor.reading = no_human and clock_sensor.reading = open_by_stranger then turn on Bosch Security Alarm and notify users via the mobile app.
- If human_sensor.reading = at_home then brightness_sensor.reading < 300 then turn on Philips Hue White and Color Ambiance.
- If co_sensor.reading > 35 then turn on Bosch Security Alarm.

Task 2: requirements thinking

Please provide 10 natural language requirements.

e.g. I hope the temperature of my room to be comfortable.

Task 3: requirements writing

Please translate the 10 natural language requirements provided in Task 2 into SH-RDL expressions.

e.g. Room_temperature should be below 30°C

Please translate the 10 natural language requirements provided in Task 2 into TAP rules.

e.g. If temperature_sensor.reading > 30°C then turn_cool Daikin Inverter AC1000.

B.2 Questionnaire for demographics information collection

Please fill in the form below.

1.What is your experiment number?

2.What is your gender?

3. What is your current age?

4. What is the highest level of education you have currently attained(or the closest equivalent)

- Less than High School
- High Schoo1
- Bachelor's
- Master's
- PhD

5. What is your major or work direction?

6. Do you have experience with programming languages such as C/C++/Java?

- Very Little
- Below Average
- Average
- Above Average

- Very Much

7. Do you have an understanding of the current technical background related to smart homes?

- Very Little
- Below Average
- Average
- Above Average
- Very Much

8. Do you have experience in using smart home systems such as Mijia and SmartThings?

- Very Little
- Below Average
- Average
- Above Average
- Very Much

9. Check whether the following statements is correct: SH-RDL contains multiple levels of requirements description.

- Yes
- No

10. Check whether the following statement is correct: In TAP programming,the object involved is a concrete device, not a certain type of device.

- Yes
- No

B.3 Questionnaire for experiment feedback

Please fill in the form below.

1.What is your experiment number?

2. Did you know the purpose of the experiment?

- Yes
- I didn't know

3. Did you fully understand the requirements for Task 1 and Task 3 in the experiment?

- I fully understood
- I basically understood
- I need assistance from the experiment staff

4. In completing Task 1, which requirement description language, SHRDL or TAP, do you think makes it easier to directly discern the user's intent?

- SHRDL
- TAP
- Both are about the same

- Neither reveals the intent

5. In completing Task 3, which requirement description language, SHRDL or TAP, do you think better expresses your smart home intentions?

- SHRDL
- TAP
- Both are about the same
- Neither is easy to use
- It depends. Sometimes SHRDL is easier to use, and sometimes TAP is easier to use

6. Do you have any questions or suggestions regarding SHRDL?

B.4 The reference information

To help participants, we list some suggested environmental attributes as shown in Table 3, and devices in Table 4 for reference.

Table 3. Suggested environmental attributes for reference

environment attribute	range
suitable air_humidity	40%-60%
suitable air_quality	<50
suitable room_temperature	18°C-25°C
forecast_temperature	
suitable brightness	300-1000lux
time	[0-24]

Table 4. Suggested devices and their functionalities for reference.

Device	Action	State	Function
Purifier	turn_on turn_off	on off	purify_air null
Light	turn_on turn_off	on off	light_up light_down
AirConditioner	turn_heat turn_cool turn_off	heat cool off	heat cool null
Curtain	open close	open close	light_up light_down
Fan	turn_on turn_off	on off	cool null
Humidifier	turn_humidify turn_dehumidify turn_off	humidify dehumidify off	humidify_air dehumidify_air null
Vacuum	turn_on turn_off	on off	clean_floor null

AUTHOR STATEMENT

We would like to declare that the work presented in this manuscript is an original research conducted by us. The research adopts a mixed-method approach, encompassing a systematic mapping study, an online survey, and a requirements interview. These methods were employed to ascertain the needs of end-users in expressing their requirements within a smart home environment. Drawing from the outcomes of our study, we have formulated a requirements description language dubbed SH-RDL (Smart Home Requirement Description Language). This language is designed to facilitate end-user programming in smart homes.

A portion of this manuscript, particularly the online survey which forms one part of the mixed method (excluding the in-depth data analysis), has been previously publicized as a 2-page poster at the 2021 IEEE 29th International Requirements Engineering Conference (RE). The citation for this publication is as follows: Bian Han, Xiaohong Chen, Zhi Jin, Lin Liu, Smart3E: Enabling End Users to Express Their Needs for Smart Homes. RE 2021: 422-423.

However, it is important to note that the manuscript in its entirety has not been published or accepted for publication elsewhere, nor is it currently under consideration for publication in any other journal.