Bellucci, A. Vianello, A., Florack, Y., Micallef, L., Jacucci, G. (2019). Augmenting objects at home through programmable sensor tokens: A design journey. *International Journal of Human-Computer Studies*, 122, pp. 211-231.

DOI: 10.1016/j.ijhcs.2018.09.002

Augmenting objects at home through programmable sensor tokens: a design journey

ANDREA BELLUCCI, Universidad Carlos III de Madrid[1]
ANDREA VIANELLO, Aalto University
YVES FLORACK, University of Helsinki
LUANA MICALLEF, Aalto University
GIULIO JACUCCI, University of Helsinki

Abstract. End-user development for the home has been gaining momentum in research. Previous works demonstrate feasibility and potential but there is a lack of analysis of the extent of technology needed and its impact on the diversity of activities that can be supported. We present a design exploration with a tangible end-user toolkit for programming smart tokens embedding different sensing technologies. Our system allows to augment physical objects with smart tags and use trigger-action programming with multiple triggers to define smart behaviors. We contribute through a field-oriented study that provided insights on (i) household's activities as emerging from people's lived experience in terms of high-level goals, their ephemerality or recurrence, and the types of triggers, actions and interactions with augmented objects, and (ii) the programmability needed for supporting desired behaviors. We conclude that, while trigger–action covers most scenarios, more advanced programming and direct interaction with physical objects spur novel uses.

Keywords: end-user programming; smart home; do-it-yourself toolkits; trigger-action programming; physical interaction

1. INTRODUCTION

Domestic activities are the result of interactions amidst different aspects of human life, e.g., work, leisure, safety, or health. Such activities cannot be considered as separate or independent, and are the reflection of inhabitants' needs that can be quick and ephemeral, such as sharing a to-do list, or more habitual and long-lived, such as being notified when the kids are back from school [Bellucci et al. 2016; Howard et al. 2007; Judge and Neustaedter 2014].

The heterogeneous activities that occur in the domestic environment and the interactions of inhabitants with objects in the household emphasize the need of technologies that can be easily adopted and re-configured by end users [Taylor and Swan 2005]. This, in turn, calls for a more human-centered approach to the design of smart home systems that is able to foster meaningful designer-inhabitants' interactions to understand the nature of domestic activities [Lim et al., 2013]. Users should be able to set up and program domestic technologies according to their desiderata, personal values, and ways of being, whether they last for a year or just for a few moments, they are serious or trivial, they are personal or shared.

In this paper, we present findings from a design exploration that shed light on what it means and what is needed to develop end-user technologies —interactive systems that can be easily programmed and reconfigured by the user— that are able to address meaningful interactions in the domestic environment. In our design journey, we developed an open-ended toolkit (T4Tags 2.0) that consists of web-connected and versatile physical tokens embedding different sensing technologies. T4Tags 2.0 allows users to augment physical objects in the home and to quickly program the behavior of tokens through trigger-action programming with multiple triggers. Tokens can be easily attached to and detached from physical objects to program smart behaviors for real-life scenarios, thus addressing heterogeneous needs, such as detecting if the children open the liquor cabinet or supporting activities of daily living, such as checking if my elderly father has taken his medicine. T4Tags 2.0

[1] Corresponding Author: abellucc@inf.uc3m.es, Universidad Carlos III de Madrid, Departámento di Informatica, Avenida de la Universidad 30, 28911, Leganés, Madrid, Spain.

provides an on-line social platform where users can share their "recipes" of programmed tokens. This allows to support users' proactive behavior and sustained engagement: users can be inspired by system usage from other families as well as adopt or tailor others' recipes for their personal needs.

We facilitated a series of workshops in four households using T4Tags 2.0 to elicit technology-supported activities at home through a user-driven ideation of scenarios. Our goal was to explore smart home scenarios as emerging from people's lived experience with an end-user technology and the collocated interaction with domestic objects. In particular, we addressed the following research questions: (i) What are the different kind of household activities that can be supported by an end-user technology that augments physical objects? And, (ii) What type and level of end-user programmability and hardware composition (triggers and actions) are needed to support desired behaviors?

Our findings provide empirical evidence that trigger-action programming, augmented with and/or composition of multiple triggers as well as the physical manipulation of augmented objects, is practical for implementing most of desired behaviors at home. We contribute to the state of the art of end-user programming for the domestic environment with: (i) a detailed account of the diversity of domestic activities that are of interest for people, in terms of high-level goals spanning across activities in households, degree of ephemerality or recurrence, and combinations of triggers and actions as well as their relationship with existing objects at home, and (ii) a field study on the practical use of trigger-action programming and tangible interaction for the domestic environment. From the lessons learned, we outline implications for the design of end-user technologies that allow users to articulately program the landscape of artifacts at home.

The remainder of the paper is organized as follows: after framing this work with respect to the state of the art (Section 2) and discussing the terminology used in this paper (Section 3), we introduce the T4Tags 2.0 toolkit through an example scenario in Section 4. We describe the research methodology in Section 5. Then, we present the results of the workshops in Section 6. Section 7 discusses the findings from our design journey. Finally, we provide conclusions and highlight potential future work in Section 8.

2. RELATED WORK

A great deal of research has been devoted to the fulfillment of the vision of a "smart home" from different perspectives, such as functional, instrumental and socio-technical [Wilson et al. 2015]. The functional and instrumental views emphasize on the technical challenges to the development of household technologies for better supporting user needs by enhancing existing services [Heath and Bell, 2006] and enabling end-user control [Chetty et al. 2008]. Functional and instrumental perspectives focus on methods for home automation, interoperability, reliability, activity recognition, sensing technologies and infrastructures [Mennicken et al. 2014]. Projects such as OpenHAB [Hosek et al. 2014] or HomeOS [Rosen et al. 2004], for instance, address connectivity, interoperability and infrastructure issues for the development of medium to large scale device ecosystems. Other projects tackle the problem of automatically detecting and labeling domestic human activities using artificial intelligence approaches [Cook et al. 2013; van Kasteren et al. 2011] or explore proactive strategies for efficient smart home automation [Mäyrä et al. 2006; Balasubramanian and Cellatoglu 2008]. However, such technology-oriented perspectives often depict the home as an aseptic environment: they are more interested in the technical setup and hardware testing, neglecting the fact that meaningful interactions in the household are configured by our convergence with technology and the physical world, in a context where precision and accuracy give the way to ambiguity and playfulness, effectiveness to leisure, and efficiency to personal nourishment [Howard et al. 2007].

In parallel with the technical demands, a vivid research interest is focusing on understanding the socio-technical landscape to identify design challenges of general computing technologies from inhabitants' perspectives [Frohlich and Kraut 2003], thus switching the focus from "task-oriented" to "being-oriented technologies" [Vetere and Feltham 2007]. Field studies of web-connected physical tokens, for instance, revealed how domestic appropriation of smart technologies affords a variety of use scenarios that are not possible to anticipate at design time [Ylirisku et al. 2013]. Other studies investigated the wide range of interactions that occur in the household with any kind of object, including non-digital ones [Crabtree and Tolmie 2016]. The variety of needs and appropriation possibilities can be ascribed to the specificity and diversity of use contexts at home that include aesthetic [Taylor and Swan 2005], social and material aspects. For this reason, designing for the smart home requires the practical involvement of the end-user in the design process to disclose technology adoption, use and appropriation, as exemplified by research on domestic appropriations of tangible tokens to web content [Lee et al. 2014], in-the-wild studies [Mennicken and Huang 2012; Jakobi et al. 2017] and the growing availability of do-it-yourself end-user toolkits [Woo and Lim 2015; Sas and Neustaedter 2017].

2.1. Do-it-yourself and end-user systems for the home

Classical smart home technologies are difficult to understand and use by non-technical inhabitants [Mennicken and Huang 2012]. Indeed, they provide technical components, such as sensors, actuators, and cables, but they require infrastructural changes and they are not designed following a holistic tactic that addresses domestic dynamics as a whole [Bellucci et al. 2016]. Do-It-Yourself (DIY) toolkits [Woo and Lim 2015; Sas and Neustaedter 2017], on the contrary, aim to lower prerequisite technical knowledge and skills and to reduce the gap between people and technology [Hwang and Hoey 2012]. Such toolkits introduce more flexible solutions for automating people's everyday routines [Woo and Lim 2015], such as the trigger-action programming approach [Ur et al. 2014; 2016]. This, in turn, allows occupants to install and program networked sensors and actuators without the help of experts and thus to develop their personalized "life-hacks" in the domestic space. DIY toolkits inherently build upon people's experience, deep knowledge of the domestic space, the different activities that take place within, conventions established among family members and their interpersonal relationships. They provide more flexible control over the technology, which has been identified as a key requirement for effective end-user customization [Hwang and Hoey 2012]. Findings from explorations of DIY practices for complex home technology configurations has also revealed that open-endedness is a desired quality for such toolkits [Sas and Neustaedter 2017]. As a downside, DIY home systems heavily rely on users' proactive participation, motivation, and engagement, which unveils the need to introduce strategies for fostering users' participation and creative input, such engaging users as much as possible during the design phase [Sas and Neustaedter 2017], as well as provide tools and functionality to support the user at use time, such as crowd-based sharing of user-generated "digital recipes" (e.g., IFTTT.com) [Newman 2006].

DIY and end-user systems appear to be a more pliable option than dedicated systems [Mennicken et al. 2014], and a recent body of work have been produced that investigates the intersection of end-user programming with the Internet of Things at home. Novel models, systems and user interfaces that allow end users to program smart behaviors at home by the composition of trigger-action rules has been proposed, as exemplified by the work of Desolda et al. [2017] and Ghiani et al. [2017]. However, experiments on end-user programming for smart behaviors at home have been conducted in artificial settings and they often involved only tech-savvy users who are not representative of the average smart home users, thus threating the ecological validity of the study [Huang and Cakmak 2015; Ghiani et al. 2017]. This approach can be useful for developing technical features of a system, and to understand how best to design an interface that provide higher degrees of expressiveness for the end users, but does not uncover how smart home technologies are used by people in their everyday life and what are their design requirements as directly informed by in-situ interactions with the landscape of existing objects and places at home. In other cases, when the design of end-users technologies was carried out together with users, the design activities were performed outside the household context [Sas and Neustaeder 2017] and/or in time-constrained workshop setups [Brich et al. 2017], therefore hindering the possibility to capture real-world interactions with a technology in context.

While previous work demonstrates feasibility and potential of end-user technologies for the home [Desolda et al. 2017, Ghiani et al. 2017, Brich et al. 2017], there is a lack of analysis of the extent of technology needed and its impact on the diversity of activities that can be supported and the integration of the technology with existing artifacts in the household. Field studies are indeed essential to provide a better user experience for domestic technologies [Woo and Lim 2015] and to design systems that are informed by users lived experience with the technology.

The design of T4Tags 2.0 draws on the emergence of end-user trigger-action programming services and models [Desolda et al. 2017, Ghiani et al. 2017] and the alleged potential of DIY smart home products. Examples from the industry are littleBits Smart Home Kit[2], Philips HUE[3] and Sense Mother[4]. The academia has been exploring the use of programmable DIY systems for supporting people's routines in the household. Examples are Pervasive interactive Programming (PiP) [Chin et al. 2006], GALLAG Strip [Lee et al. 2013] and AutoHAN [Blackwell and Hague 2001].

The aforementioned tools show opportunities for adopting a tangible approach to end-user programming of the domestic environment. For instance, PiP and GALLAG Strip adopts a programming-by-demonstration approach [Hartmann et al. 2007]. PiP allows users to program the behavior of networked devices by physically interacting with them or by defining their behavior through a graphical interface (PipView). GALLAG Strip offers instead a series of magnetic and motion sensors that can be attached and detached to existing object, e.g. for sensing the opening and closing of a door or detecting the presence of a person inside a room. It also provides smart plugs that can sense the state of electric appliances. AutoHAN provides tangible direct manipulation of cube-like physical devices that acts as remote controllers for domestic appliances through buttons and other interfaces such as IR transceivers. A more recement example of the tangible programming paradigm for the home is the Improv system [Chen and Li 2017], which exploits programming-by-demonstration to support end-users in composing on-the-fly cross-device interfaces to an existing application. Finally, Sas and Neustadter [2017] comment that tangible programming is a promising paradigm to pursue in order to support more engaging interactions with DIY electronics. Our approach uses tokens that embed sensors and actuators that can be used as physical labels and attached to objects at home. Domestic objects, therefore, are augmented with sensing technologies and they can be used as triggers

[2] http://littlebits.cc/kits/smart-home-kit
[3] http://www2.meethue.com/
[4] https://sen.se/mother/

for smart behaviors; for example by detecting the presence of an object in a room or the movement that results from the user interacting with the object.

2.2. Trigger-action programming

A wealth of recent research [De Russis and Corno 2015; Ur et al. 2014; Huang and Cakmak 2015; Ur et al. 2016] suggests that the trigger-action format is the most exploited approach for end-user programming since people without programming experience are able to create recipes to self-configure the behavior of smart devices. However, simple trigger-action programming like the one afforded by IFTTT.com that relies on basic rules, is not enough to program smart behaviors. Even if many actions in the smart home can be expressed as a result of a single trigger event (e.g., "turn on the light at 6.00 pm"), 22% of desired behaviors required the composition of more than one trigger or action (e.g., "turn on the light at 6.00 pm and I am at home") [Ur et al. 2014]. Brich et al. [2017] conducted a contextual inquiry with 12 families on home automation needs and confirmed that simple trigger-action notation is suitable for the definition of straightforward automation tasks but it does not provide the flexibility needed to support more complex scenarios. Other works from the state of the art suggest that complex behaviors can thus be supported by extending trigger-action programming for including conditional operators and constraints [Mennicken et al. 2014; Desolda et al. 2017; Ghiani et al. 2017]. An often-cited approach is the one implemented by iCAP [Dey et al. 2006], which enriches trigger-based programming with contextual and environmental events, such as spatial, temporal and personal relationships.

Allowing richer expressiveness, however, does not come without problems: the composition of conceptually different triggers and action types, in fact, can cause ambiguities in the user interpretation of a result [Huang and Cakmak 2015]. Ghiani et al. [2017] presented and end-user system that implements an interface to program composition of trigger-action rules with a distinction between events, conditions and different types of actions (e.g., actions that reverts back to their original state automatically versus sustained actions or states). In this way, they aim to reduce misinterpretation and support easy definition of complex rules. Their study reinforced previous findings from the literature [Huang and Cakmak 2015], showing that augmenting the expressive power of the language to define the rules results in more complex behavior being programmed by the users. The study also highlighted the need to investigate more "natural" approaches to program advanced trigger-action behaviors. As a matter of fact, the authors proposed a description of the rule in a "natural language" that made use of "when," "if," and "do" keywords. However, the participants in the usability study did not find the language to be natural at all, and this outcome points to the need to find other strategies to support end users in the definition and understanding of rules.

Successful trigger-action applications stress on the need to provide visual metaphors, to engage inhabitants that are reluctant towards to the use of technology at home, and tangible approaches, such as programming sensors by physical demonstration, to make it easier for the users to materialize their needs without textual programming [De Russis and Corno 2015]. Metaxas and Markopoulus [2017] showed that enriching logical operators with semantic information describing the range of outputs from available services as well as using contextual information to minimize logical expressions allow end users to compose complex rules more efficiently, limiting the numbers of logical errors. Real world tangibility [Lee et al., 2013] is another key factor to take into account for smart home end-user toolkits: allowing the user to interact with real world objects while programming their behavior can leverage the familiarity with everyday objects in the household to elicit purposeful actions.

Previous work highlights that there is a need of field studies to gain a better understanding on the practical adoption of trigger-action programming in the domestic environment. Mechanical Turk studies [Ur et al. 2014; Huang and Cakmak 2015] using artificially-generated tasks and laboratory experiments [Ghiani et al. 2017] are not able to capture meaningful behaviors from the real-world experience of the participants.

3. TERMINOLOGY

Throughout this paper we consider *Do-It-Yourself* (DIY) *toolkits* to be systems that provide a series of sensors, actuators, and software services that can be easily installed and configured by the end users. In the case of smart homes, by introducing versatile programmable sensors, they allow the quick development of solutions that adapt to the complexity of different households and their constantly changing needs. We focus on the users experience of integrating readily available products (such as smart bulbs, wireless sockets and sensors) into the landscape of objects at home, rather than exploring DIY practices of building smart home hardware at circuit level, as investigated by Sas and Neustaedter [2017].

Trigger-action programming is a simple programming paradigm that allows users to define interactive behaviors as production rules in the form *if something happens* (the trigger), *then do something* (the action) [Huang and Cakmak 2015; Ur et al. 2014]. A popular example of the trigger-action programming paradigm is offered by the online service IFTTT.com, which exploit trigger-action rules to mashup different web channels and devices.

We refer to a *digital recipe* (or simply *recipe*) as the description of a desired functionality, behavior, or activity in terms of the composition of input/output data and control statements. Digital recipes are like culinary recipes: input/output data are the ingredients, while control statements are the instructions on how to combine the ingredients to have the desired outcome (the dish). Digital recipes support the end-user composition of multiple configurations. Users can change, for instance, an input source in one recipe to obtain a different result, or explore different configurations that realize the same outcome. Such configurations can be shared with others in order to help and inspire other users with similar needs [Newman 2006]. In our system, a recipe is defined as one trigger-action production rule with multiple triggers combined via and/or operators. An example of a simple trigger-action recipe could be "if it is 6pm, then turn on the light".

Figure 1: The T4Tags 2.0 toolkit.

4. T4TAGS 2.0

T4Tags 2.0 [Bellucci et al. 2016] is a ready-to-use toolkit that includes (
Figure 1):

- Four Bluetooth Low Energy (BLE) physical tokens embedding different sensing technologies: a physical button, Near-Field Communication (NFC), motion, light and temperature sensors. The technology is built around a LightBlue Bean[5] Arduino microcontroller and, together with a magnet, is enclosed in a 3D-printed case that offers a blackboard material to (re)write on the front of the case;

[5] https://punchthrough.com/bean

- A set of 18 double sided re-stick-able (washable) adhesive squares for attaching/detaching the tokens to different objects or surfaces;
- Two Philips Hue[6] smart light bulbs;
- Two Fibaro Wall Plug[7] smart power outlets;
- A wireless base station that encloses a Raspberry PI[8] device with a NFC reader, and BLE, Z-Wave, Zigbee and Wi-Fi connectivity.

The station is incorporated into a wooden case, and together with a Web-based application, allows to program the behavior of tokens, smart bulbs, and power sockets. Tokens have different colors (pink, blue, green and yellow) to make them easier for the user to recognize and program. The system interface, in fact, uses the same color to identify the token, which allows the user to know how a specific token was programmed. Light bulbs and power outlets also have different colors to identify them (pink and yellow). A dedicated application that can be installed on personal mobile devices allows to receive different kinds of notifications (e.g., sounds).

Figure 2: The user interface of T4Tags 2.0 to create "recipes" for the tokens. On the left the main interface and on the right two examples of AND/OR compositions.

Figure 3: The "Recipe browser" page that shows the recipes created by the user ("Your Recipes" section), and those created and shared by others ("Public Recipes" section).

Figure 2 shows the T4Tags 2.0 user interface to program physical sensor tokens to define a recipe that implements a desired smart behavior. Users get to this screen after they clicked on the button to create a new recipe as showed in Figure 3 (the "plus" button on the bottom right corner). Figure 3 shows the "Recipe Browser," which is the main screen of the application and provides a list of recipes created by the user or shared with others. We followed Material Design[9] guidelines and provide the "plus" button to perform the primary, or most common action on the main screen: that is, creating a new recipe. When creating a new recipe, users can assign pictures to the recipe by adding existing pictures or taking photos with the camera of their tablet or mobile phone, they can add a description to the recipe and select the privacy level of the recipe: private, shared with family members or shared with everybody (Figure 2, left, top panel). The middle panel of the user interface in Figure 2 (left) shows the programming interface: here users can define the composition of trigger-action rules by selecting among six different triggers (temperature, light, scan NFC tag, motion detection, timed events, button) and five actions (send an email, play a sound, turn on/off light, turn on/off power outlets, manage lights color and brightness). The bottom panel of the user interface provide feedback to the user of the token that is being programmed (the token that the user has put on the wooden tray) and also information on the output; for example, if the user had chosen to turn on a power outlet, the interface would show the selected power outlet and its state.

Through its authoring user interface, T4Tags 2.0 allows the composition of different triggers with conjunction (AND) and disjunction (OR) logical operators to express more complex behaviors. We designed an interface that allows the user to visually select AND/OR operators as intersection or disjunction of two different triggers (Figure 2, right). The top-right production rule in Figure 2 implements the following behavior: "Turn on the light when I wake up." To this end, the user implemented the conjunction of two conditions to recognize when he wakes up:

[6] http://www2.meethue.com/en-US
[7] http://www.fibaro.com/en/the-fibaro-system/wall-plug
[8] https://www.raspberrypi.org/
[9] https://material.io/design/components/buttons-floating-action-button.html#usage

shaking a token (thus performing a gestures) and the lights in the room are off (this state is monitored through the light sensor in the token). If both conditions hold true, the lights in the room will be turned on (using a smart light bulb). As shown in Figure 2, the user had chosen the green token for both triggers. The bottom-right production rule in Figure 2 implements the following behavior: "if it is hot or there is too much sun in my room, turn on the fan." To do so, the user implemented a disjunction of triggers: he programmed the blue token to monitor the temperature and defined a trigger event when the temperature goes above 24 degrees. He then programmed the same token to monitor the illuminance of the room (again using the light sensor embedded in the blue token). When one of the two triggers is activated, the yellow power socket is turned on, thus activating the fan that was previously connected to it.

While augmenting the expressive power of the rules [Ghiani et al. 2017] and potentially support a wide range of domestic scenarios, the combination of triggers can cause ambiguities, as already uncovered by previous research [Huang and Cakmak 2015]. For instance, the conjunction (AND) of two triggers might lead to situations that are never evaluated as True, e.g., "if the temperature is above 30° and the temperature is below 20°," or that are very unlikely to happen "if the doorbell rings and it is 11:00." Similarly, disjunction (OR) can lead to situations that are always evaluated as True: "if the temperature is above 25° or the temperature is below 30°." We chose to use a conservative approach to cope with this problem as suggested by Huang and Cakmak [2015], and disallow confusing options by restricting the choice of the triggers that can be combined through conjunction and disjunction, according to the type of triggers and the conditional operator. This approach does not restrict the expressiveness of the toolkit and implements a path of least resistance, which has been demonstrated to be a desired component of successful toolkits [Myers et al. 2000]: the toolkit guides the user toward to implementing correct behavior and avoids the creation of rules that lead to inconsistent or invalid states.

The toolkit supports the distinction between two types of triggers:

- *Event triggers*: instantaneous events or transitions from one state to the other, e.g., "a motion is detected" or "the temperature *goes* above 30°." For defining event triggers, the interface uses words such as "*goes*", "*drops*";
- *State (or condition) triggers*: conditions that can be evaluated as True or False at any time, e.g., "if it is between 5.00 – 7.00 pm or the light is dim".

The toolkit implements two type of actions:

- *Instantaneous actions:* that execute a specific task in a single step (like sending an email or playing a sound). Instantaneous actions do not change the state of the system.
- *Sustained actions:* changes of the state of a device (e.g., a light bulb or a power socket) that do not revert back automatically to its previous state

We did not implement *extended* actions —actions that automatically revert to the original state when they end— in this version of the toolkit.

Two triggers can be combined with a conjunction operator only if they are both state triggers, or if one is an event trigger and the other is a state trigger. The interface does not allow conjunction composition of two event triggers, thus avoiding the possibility to create rules that are extremely unlikely to occur. To provide an example, the toolkit allows to express rules such as "if the temperature is above 15° and the temperature is 20° or below" because they are both two state triggers. Considering for example the case in which the current temperature is 21°. The first operand of the conjunction is True, and the second operand is False. Thus, the rule is evaluated as False. If the temperature drops from 21° to 20°, the first operand remains in the same state (True) and the second operand transition from False to True: the rule is now evaluated as True. Lastly, if the conjunction or disjunction of two triggers results in a rule that is always evaluated as True or False, the system will provide a prompt to the user, informing that the specific behavior cannot be implemented, explaining the cause. We plan to introduce suggestions that would guide users in correctly implement rules in a future iteration of the toolkit.

4.1. Creating a recipe for the domestic environment with T4Tags 2.0

We present a walkthrough of a real-world scenario to introduce the overall functionality and possibilities offered by the T4Tags 2.0 toolkit for programming smart home behaviors.

Figure 4: A sample scenario for the kitchen. (a) A token is attached to the milk box. (b) The token senses the milk box temperature when left on the kitchen counter and blinks or play a sound when it reaches 20 degrees. (c) Users are reminded to put the milk box back in the fridge.

A family wants to remember to put back the milk box in the fridge after breakfast (Figure 4). To this end, the mother taps the "if" button and selects the icon that represents the "temperature" trigger (bottom-most icon of Figure 5a). Then, among the available options —the system provides three icons representing "above", "below" and "range"— she chooses the "above" trigger to check when the token reaches the room temperature (e.g., 20°).

Figure 5: The user interface to select triggers. (a) Available triggers. (b) Available options for the trigger "temperature." (c) User interface for the definition of the temperature.

After having saved the trigger options, the mother selects the action she wants to be performed. She taps the "then" button that appeared after the trigger has been chosen, and then selects the "play sound" action among the available ones (bottom-most icon of Figure 6a). Also in this case, she is able to define action-specific options, such as the sound she wants to be played. She also selects the device where she wants to receive the notification, e.g., her smartphone that she previously associated with the system. In this way, the audio feedback will be activated in her smartphone for catching her attention and thus be reminded to put the milk back in the fridge. At this point, the interface invites her to place a token on the wooden case to assign the corresponding behavior to the token and finalize the recipe (Figure 6c).

Figure 6: The user interface to select actions. (a) Available actions. (b) Available options for the action "play sound." (c) The system invites the user to place a token on the wooden case to assign the recipe to it.

Before saving the recipe, the mother uploads a picture to provide a visual glimpse of the goal of the recipe (Figure 7). The family can find the newly generated recipe in the "Recipes browser" page (Figure 3, "Your Recipes" section).

Figure 7. Picture uploaded by the mother to better describe the behavior of the recipe.

4.2. Crowd-based recipe sharing

T4Tags 2.0 also provides a web-based social platform to share implemented recipes. The recipe sharing functionality has been developed to provide a platform to browse, explore and take inspiration from other users' examples and implementations. The "Recipe Browser" page of the application (Figure 3) is organized into two sections:

- The “Your Recipes” section, that shows the recipes created or adopted by the family, which can be enabled or disabled (on/off switch in the top-left corner of Figure 3);
- The “Public Recipes” section, that shows the recipes created by other users, which can be explored and used as-is or modified according to specific needs. Recipes can be filtered according to high-level categories as identified from the results of workshops with families (e.g., safety, parenting, entertainment) or by the trigger technology.

Each recipe is presented as a rectangular panel split into three horizontal parts. The first part is the pictorial representation of the recipe or its category. It can be either a photograph showing a usage example of the recipe taken by the user, or an icon representing the category for the recipe. The second part includes the recipe name, and the maker of the recipe, e.g., “by John”, in case the recipe is in the public library. The third part displays the functional aspect of the recipe, representing the general trigger(s) and action with a directed arrow between them to show the order of events. In case of recipes in the personal library, the trigger(s) and action are also color-coded with the color of the corresponding token or device, e.g., if the pink token is responsible for measuring the temperature, the trigger will show an icon of a thermometer and have a general pink background; if the yellow plug turns the electric fan on, the action will show an icon of a plug and have a general yellow background. Public recipes are color-coded grey in the visual functional description as no devices have been linked to them before adopting them into the personal library. In cases when triggers or actions are not linked to physical devices, the color remains grey (e.g., for digital services such as time or messaging).

5. RESEARCH METHODOLOGY: DESIGNING FOR, WITH, AND BY THE USERS

Previous work is characterized by a scarcity of in-situ deployment to elicit relevant real-life needs for trigger-action programming [Huang and Cakmak 2015]. It is difficult, therefore, to understand how people would actually program domestic artifacts in the real context of use [Ur et al. 2014; Ur et al., 2014] or how their interaction with a domestic technology would change people’s perception of it, depending on the benefits and drawbacks of the technology itself [Brich et al. 2017]. As pointed out by Huang and Cakmak [2015], for instance, the drawback of having many users and potential activities in a Mechanical Turk study is that participants worked with programs that were not the direct result of their ideas or situated needs. Brich et al. [2017] also reported that a major limitation of their study of the domestic environment, which adopted contextual inquiry and did not involve the deployment and use of a real home automation technology, is that participants “merely thought about potential use cases rather than using real-world home automation,” because they didn’t have a first-hand experience of what is like to live with such technologies. In the case of field studies with DIY toolkits deployed in situ, however, such as the study conducted by Woo and Lim [2015], the major issue is that the technology was treated as given: the user had limited possibility to be reconfigured in order to explore heterogeneous uses and appropriations.

In our research, we tackle these issues by adopting a multi-techniques approach of designing with and by the users with in situ deployment of domestic technologies. We employed contextual inquiry [Beyer and Holtzblatt 1997] together with the deployment of technology probes [Hutchinson et al. 2003] at early stages of design, with the goal of developing a rich and grounded understanding from the field about how people use and integrate technology with their existing activities and routines. Our overarching design approach is grounded on the research through design model for interaction design research [Zimmermann et al. 2007]. Conversely from other approaches to develop home technologies, for instance the work of Brich et al. [2017] that focused on understanding the expressiveness of rule- and process-based notations for home automation using a tool- and interface-independent approach — use cases were implemented on a piece of paper— we were deliberately biased toward to what to consider. After an initial stage of design with a technology probe, we developed the T4Tags 2.0 prototype that we used as a "tool to think with" the user [Papert 1980], giving us the possibility to explore limitations and benefits of our DIY approach for home technology, that is, augmenting physical objects with programmable sensor tokens. Our approach resonates with the design methodology to develop ubiquitous home technology proposed by Schmidt et al. [2007]. They showed that presenting prototypes and having the users interacting with them in the domestic context stimulates users creativity and allows to capture more nuanced design ideas.

Being biased toward to the use of a digital tool has the limitation of not capturing all the mental models and configurations that users are likely to come up with pen-and-paper studies [Brich et al. 2017]. We, however, acknowledge that the tools we use shape the way we think [Turkle 2017]. According to the theory of embodied cognition "interacting with tools changes the way we think and perceive" [Kirsh 2013], therefore also using a pen-and-paper approach introduces constraints to the final outcome. In our investigation we deliberately choose to follow a designerly approach to produce knowledge through the design of an artifact that reflects our specific framing of the smart home problem. In doing so, we materialize our design ideas into an interactive system, which is the result of our design journey with the users, and we situate this system within the multitude of other systems that have been created to address the same common problem from different perspectives. This, ultimately, would allow the design research community to analyze, compare and critique this pool of different artifacts in search of general implications and design patterns [Zimmerman et al. 2007].

Step 1: Probing the domestic environment. The process started with a formative study —a one-month deployment in a household— in which we used a technology probe approach [Hutchinson et al. 2003] to inspire discussion in the context of people's home and identify users' needs for generic technologies and platforms for the domestic environment. We aimed to uncover affordances and emerging designs as revealed by the family lived experience with the technology.

The technology probe, the T4Tags 1.0 toolkit (see Figure 8), was built upon the concept of Tokens of Search [Lee et al. 2014] and provided a system to easily link web content to physical objects via NFC. T4Tags 1.0 consisted of:

- a set of 3D-printed physical tokens with different shapes, colors and affordances with embedded NFC tags and NFC stickers;
- a web server that stored the content of the physical tokens;
- a web interface that allowed to edit the content of a token by adding or removing (drag and drop) web links;
- a tray that embedded a WiFi-connected device with a NFC reader that was used, in combination with the web interface, to edit the content of a token;
- an Android mobile phone that run a dedicated application to retrieve and display the content of a token;

Figure 8: The T4Tags 1.0 open toolkit.

T4Tags 1.0 implemented the core functionality, informed by findings of previous studies [Lee et al. 2014], to allow users to associate any number of web URLs to a physical token. In order to attach digital to content to the physical tokens, users first put the token they want to use over the tray. The reader retrieves the ID of the token together with the associated content. The users, then, can add or remove content by dragging and dropping URL addresses from a browser to the window of the web interface. Users can then retrieve the content of a token by using a mobile phone with an installed Android application that recognizes the NFC tag and displays the list of associated URLs. Figure 9 shows a scenario that can be implemented with the system: a token can be used to store and subsequently retrieve heterogenous content, in the form of a URL, about a family trip (hotel bookings, flights or images of locations to visit).

Figure 9: The implementation of a scenario. A token is used to store information to plan a family trip: hotel booking, flights and images of different locations to visit.

The initial phase was based on rapid cycles of use, design, development and in-situ deployment of the evolving prototype. A family lived with the toolkit for one month and the researchers carried out weekly co-design sessions to capture a reflective account of various daily situations as articulated by participants. Co-design was inscribed in Extreme Programming activities [Bellucci et al. 2015]: an agile development model that gives prominence to the rapid availability of usable prototypes to accelerate the exploration of the design space through rapid cycles of software releases. Each week the researchers would identify with the users the functionality to implement for further development of the initial prototype. The in-situ experimentation allowed an extensive exploration and experiencing of appropriation scenarios that led to the identification of core software and hardware features for a next iteration of the system.

Step 2: Designing an end-user DIY toolkit. From the findings of the previous step we identified four design goals that drove the design of T4Tags 2.0 as a DIY domestic toolkit intended to provide (i) a repurposable technology (domestic technologies need to be pliable so that can be easily reconfigured to address emerging needs instead of being discarded when the service they originally provided is no longer of use), (ii) end-user programming functionality that upholds the needs for advanced programmability of domestic technologies by nonprogrammers as well as easy integration and recombination of different sensing technologies, which in turn support the quick composition of hardware and software building blocks for heterogeneous uses, (iii) crowd-fueled sustained appropriation to spur novel uses and support sustained user engagement, and (iv) sensors that could be easily integrated into the domestic space, augmenting existing artifacts and providing different input/output functionality for tangible interaction.

Step 3: In-situ user study for user-driven ideation of scenarios. We used T4Tags 2.0 to run an in-situ user study —full day workshops with four families— to gather broad descriptive data on what types of routines can be supported by trigger-action programming and tangible sensor tags, and what level of end-user programmability is needed. We wanted to generate and collect usage scenarios as well as observe how people use the toolkit to create programs at home that are the result of their real-world needs and are informed by their knowledge of the domestic environment. In these workshops, we were able to explore issues for the design of domestic end-user systems such as: (i) technological integration with the landscape of artifacts at home, (ii) the new possibilities that were opened up by extending basic trigger-action programmability with and/or compositions of trigger/action events when combined with programmable physical tokens, and (iii) how sharing recipes helps users helping themselves in the creation of smart behaviors.

In this paper, we report on the third step of our research. A detailed account of the formative study (step 1) can be found in Bellucci et al. [2015], while the design goals (step 2) are presented in Bellucci et al. [2016].

Table 1. Details of the families involved in the four workshops.

6. IN-SITU STUDY: USER-DRIVEN IDEATION OF SCENARIOS

Four Finnish households were recruited to participate in a one-day (8 hours) workshop at their homes to elicit usage scenarios. The families consisted of young parents (age ranged from 37 to 41 years) and two/three children (age ranged from 4 to 16), for a total of 17 participants. The variety of gender and ages among family members allowed us to observe which kind of social interactions are supported by our DIY toolkit. All parents speak fluently both English and Finnish, while most of the younger children speak only Finnish. Details of the families are provided in Table 1. The families were recruited among the network of acquaintances of the researchers. We selected families with no previous experience with smart home apparatus or DIY smart home toolkits. Families were compensated with a movie ticket for each member.

6.1. Procedure

At the beginning of the workshop, the researchers —the same two researchers conducted all the workshops— briefly introduced the aim of the study and interviewed the family to collect demographic data and information about their living situation, family life and technology use (30 minutes).

After having deployed T4Tags 2.0 in the household (connecting the base station to the home network), participants were introduced to its main functionality and they were shown how to create a sample recipe, e.g., send an email when pressing the button of a token (20 minutes). Then, a creativity warm-up (10 minutes) took place aimed at stimulating family members' imagination for the brainstorming session that followed. During the brainstorm (1 hour), which was facilitated by one researcher, each family came up with different scenarios and ideas of desired usages of the toolkit, which could also include technology not currently offered by T4Tags 2.0. Such scenarios were captured on paper notes by the another researcher.

After the brainstorming session, participants created recipes of their ideas using the T4Tags 2.0 and then tried out the implemented recipes (4h). Participants were instructed to use the available tokens, light bulbs, power outlets and tablets, and to ask the researchers if they needed any support. The researchers observed participants while interacting with the system and step in only when asked or when participants get stuck. In that case, the researchers started a conversation with the participants, trying to understand from them what was the cause that impeded the implementation of the recipe (e.g., they were not able to find a trigger-action combination for the desired behavior). The second, third and fourth families could also browse the recipe library that was populated with implemented scenarios from the other families.

Participants also acted out those ideas that could not be implemented (e.g., required sensors were not available or the scenario expected to hack domestic appliances). In this case participants were asked to act as if they had the material (e.g., sensors or features of the system) and to provide an interaction walkthrough to demonstrate to the researchers what they would do to implement the recipes and how they would work. To support such process, participants were provided with a set of paper-based mockups of different sensors that they could easily attach to the toolkits' components or to any other artifact. They were also provided with blank mockups on which they could sketch other desired technology (see Figure 10).

Figure 10: (a) Paper-based mockups of sensors and actuators that were provided to participants. They were also given blank mockups on which they could sketch desired technology. (b) Participants extended a token with a "water-level" sensor (they used the available mockup for a moisture sensor) and attached it to a bathtub for measuring the level of water while filling it. (c) Participants simulated the implementation of a recipe for such scenario: the "water-level" trigger was replaced with the available "temperature" trigger.

Participants could come up with new ideas during the implementation, try out or acting out phases. As a matter of fact, the brainstorming session served mainly to produce seed scenarios to engage participants with the active experimentation of the toolkit. During the workshops, many new scenarios were envisaged only after having used the technology. At the beginning, participants were reluctant to think about possible usages. One participant said: "*physically interacting with stuff sometimes is thought-provoking, thinking isn't always as thought provoking*." Therefore, many of the scenarios arose during the actual use of the system in the context of their home, rather than from the brainstorming. This is in line from the findings from previous research that recognizes that people often do not feel they need technology unless they use it [Schmidt et al. 2007] and that users are inspired in the creation of new routines when using technology to solve their current problems [Bellucci et al. 2014]. All the workshops ended with a brief interview (30 minutes), aimed at gathering information about parents' and children experience with the system.

6.2. Data collection and results

All the phases of the workshops, including the brainstorming session and the interviews, were done in English and video recorded. During the four workshops, we were able to collect and document a total of 111 ideas. Such ideas were collated and cross-referenced: the result was the identification of 100 unique scenarios. Usage scenarios were then coded according to distributed cognition descriptive framework [Hollan et al. 2000]. This resulted in a description of scenarios in terms of:

- The *creator* of the scenario;
- *Entities,* i.e. the persons or objects involved in the scenario. For example, in the scenario "switch on lights when I enter the home", the entities are: (a) family member, i.e. the person that enters the home, (b) motion sensor, i.e. the sensor that senses the person entering the home, and (c) lights, i.e. the object of the action;
- The *locations* where the scenario may occur, e.g., the living room or the kitchen, and if the scenario is (a) *static*, e.g., it takes place in a single location, or (b) *nomadic*, it involves a change of location from one room to another;
- *Ephemerality* or whether the scenario describes a short or a more permanent, long-lasting activity in users' everyday life. Values can be (i) *daily*, (ii) *weekly*, (iii) *monthly*, or (iv) *occasionally*. We also keep track if the activity occurs more than one time, e.g., three times a day, and consider activities that occur only in specific seasons;
- *Feasibility*, which defines if the scenario is (i) *readily implementable*, (ii) it needs *minor hardware/software extensions* of the toolkit, or (iii) *not feasible*;
- *Programmability* or the type of end-user programming that is necessary to implement the scenario, which can be (i) *basic trigger-action* programming, (ii) *advanced trigger-action* programming, or (iii) *specialized programming*;

Table 2. Categorization of scenarios according to their high-level goal[10].

Scenarios were categorized into 11 high-level goals: *Comfort, Safety, Parenting, Resource Conservation, Health & Fitness, Home Maintenance, Object Reminder, Entertainment, Security, Cooking* and *Education*. We discuss high-level goals, ephemerality, feasibility and programmability in more detail in the following sections. We also report on the integration of the end-user technology with the domestic context and existing artifacts.

6.2.1. Types of domestic activities and their technical feasibility

Table 2 summarizes the categorization of the domestic activities (the scenarios) according to high-level goals. Most scenarios were generated in the *Comfort* category, which includes those tasks to guarantee or increase inhabitants physical comfort, e.g., automatically turn on the airflow control when it is too hot, as well as preventing annoying or disturbing situations, e.g., turn off the fire alarms' blinking lights during night. The different categories indicate that the in-situ elicitation study was able to capture and describe the heterogeneity of activities that can happen in the domestic environment, which range from those related to users' comfort or entertainment, to those related to more critical human needs, such as safety, and parenting.

Examples of scenarios include being notified to pick up the keys and/or the smartphone when leaving the home (*Object Reminder*). Such scenario was partially implemented by the first family. They created a recipe with the toolkit and then attached one token to the key ring (Figure 11a) and one to their house's main door (Figure 11b). Then, one of the participants left the house without having the keys with her: the token attached to the door was programmed to sense when someone opens the door (by sensing a custom motion) and to check if the token attached to the key ring is near (proximity sensing). If the token is not in the proximity of the door, the system notifies the user by blinking and/or playing a sound on the user's mobile phone (Figure 11c).

Figure 11. Partial implementation of a scenario for being notified to pick up the keys and/or the smartphone when leaving the house (Safety high-level goal). (a) Attaching the token to the key ring. (b) Attaching the token to the main door. (c) Opening of the door and triggering a notification.

The scenario, however, is overly simplistic: people don't want to check for the keys any time someone opens the door, but only when the owner of the keys is leaving the home. To this end, the recipe should have been modified to exploit the user's smartphone —previously registered to the T4Tags 2.0 application— as proxy for identity, and check for the proximity of the smartphone to the door. The final production rule generated by using the toolkit's user interface would be (this more complex rule was implemented by the researchers):

```
IF
trigger.token_1.motion_1                        AND
proximity(token_1, token_2) == far              AND
proximity(token_1, smartphone_1) == nearby

THEN
smartphone_1.beep()
```

[10] The table reports the frequency of scenarios per category and the overall feasibility of each category. Examples of scenarios that have been acted out (A) or implemented (I) by the four families are provided. In the *Feasibility* column, the visualization shows the percentage of scenarios for each category that: are readily implementable (R), need minor hardware/software adaptations (H/S) or, cannot be implemented (N).

considering *token_1* to be the id of the token attached to the door, *token_2* the token on the key ring, *smartphone_1* the user's smartphone, and *motion_1* the custom motion to capture the door opening behavior.

Figure 12: Participants acting out a scenario for automatically switching on and off the veranda light. (a) Attaching the token to the veranda door. (b) Opening and (c) closing the veranda door.

Considering the *Resource Conservation* category, the third family implemented a recipe for automatically switching on and off the veranda's light when they open and close its door, respectively. To this end, they attached one token to the veranda door (Figure 12a) and created a recipe by defining a custom motion for switching on and off a power plug (and the connected veranda lamp) when the door has been opened (Figure 12b) or closed (Figure 12c) by a family member.

Table 2 also reports the feasibility of scenario per category, that is, how viable is the implementation of the scenario using the current technology provided by T4Tags 2.0. The feasibility levels are:

1. *Readily implementable* (R): the scenario can be implemented with the hardware/software technology currently provided by the system;
2. *Hardware/software extension* (HS): the scenario could be implemented by extending the components of the toolkit, e.g., adding a new sensor or software service.
3. *Not Feasible* (N): the scenario requires changes in the hardware or software of the domestic environment such as hacking existing appliances, or is too far-fetched.

Overall 37% of the scenarios were readily implementable, 42% needed extension of the hardware/software functionalities and 21% were not suitable for implementation with the toolkit. Scenarios that demanded for an extension of the functionality of the toolkit required to use traditional programming that could have been easily implemented provided the integration of existing online applications and services such as Twitter or Facebook. With respect to the hardware, such scenarios needed the inclusion of new sensors in the system. For instance, a scenario in the *Home Maintenance* category needed a moisture sensor for automatic watering the plants while one in the Comfort category needed a linear actuator to "open the window when I come home in summer."

The *Hardware/Software extension* dimension allows designers/developers to reflect on the efforts that are needed to design and implement new functionality to support desired behaviors. Including a new sensor, for example, does not require big implementation efforts but it needs to take design and engineering choices such as integrating the new sensor in the existing token design or, instead, providing new modules. The current implementation of the toolkit allows to send emails exploiting the Gmail APIs. New channels can be seamlessly integrated if they offer APIs for developers: this would allow to extend the space of possible triggers and actions combinations (e.g., "post a message on Twitter" or "update my status on Facebook"). The scenarios coded as *not feasible* were either too far-fetched (e.g., "automatically detected when I am short of coffee capsules and order a new batch from Amazon") or they needed to extend the hardware and software capabilities of existing domestic appliances, e.g., "programming the coffee machine to brew my favorite coffee when I press the token's button." While scenarios in the *Hardware/Software extension* were not attained because of a current limitation of the prototype, such as the lack of support for a specific service such as Twitter or Facebook, which over time can be solved by integrating the service through its public API, the *not feasible* category reveals either limitations of the trigger-action programming paradigm or the design choices that do not make impossible to implement the scenario. This category captures the feasibility of the scenarios from the perspective of the end user, given the current domestic context of the households in our study. According to previous findings [Ur et al. 2014], scenarios that could not be implemented by an end-user trigger-action approach would involve specialized functions. However, the complexity was not found in the programming logic but instead in the dedicated technology and programming effort that would be required to implement the scenario. For instance, the complexity of "automatically detected when I am short of coffee capsules" resided for the participant in identifying a sensor that she can use to program the concept "be short of" into a computable trigger event. A load cell could be a solution and a user could exploit a programming-by-demonstration approach to instruct the system that "short of" means when there are three capsules left. This can be achieved by putting three capsule in a box equipped with a load cell and letting the system to register the weight, that can be used later as a trigger. Another group of scenarios in such category required instead to hack the hardware/software of an existing device, such as modifying the normal behavior of the drinking machine to make a hot or cold drink based on the outside temperature. This kind of scenarios are not suitable for end-user repurposing. If future domestic appliances were designed in order to expose services and functionalities through common Application Programming Interfeces, they can be integrated as new channels in the toolkit, as it is already happening with trigger-action web services such as IFTTT.

As expected, the elicitation study uncovered a wide variety of needs and appropriation possibilities that are ascribed to the heterogeneity of domestic activities and the ways in which inhabitants continually re-arrange and integrate informational artefacts to support such activities. Figure 13 shows the scenarios organized according to their *ephemerality*, from activities that are occasional and eventually occur only once in a lifetime, to more permanent and recurring routines that can happen also more than once a day. The chart categorizes household activities into four dimensions (Seasonal, Occurrence, Frequency and Time Period). For each dimension, a horizontal bar is shown for each of its possible value of that dimension. The width of the bar denotes the absolute number of matches for that value. Starting with the first dimension (Seasonal), each of its values ("no" and "yes") is connected to a number of values in the next dimension, showing how that value is subdivided. The first dimension shows the *seasonality* of domestic activities, that is, variations in activities that occur at specific seasons (e.g., activities that occur only in winter versus activities that occur only during the summer). We defined a binary value for this dimension, according to whether the activity has a seasonal character or not ("no" is represented by blue lines in the chart and "yes" by orange lines). Then, activities are categorized according to their *occurrence*, which defines a time span for the activity that can be *daily*, *weekly*, *monthly* or *occasional* (for activities that occur sporadically or only once in a lifetime). We also took into account the *frequency* of the activities and considered whether they occur or can occur *once*, *twice*, or *undetermined* (potentially several times) during their time span. Lastly, we considered the *time period* when an activity occur in its time span, which can be *morning*, *afternoon*, *evening*, *night*, *all day long*, or *at some point of the day*.

Figure 13: Ephemerality of activities. The parallel set chart shows scenarios' frequency and occurrence. Blue lines show activities that are not seasonal, while red line are used for seasonal activities.

Figure 13 shows examples of scenarios according to the different dimensions. An example of a seasonal scenario is "eating the summer cottage for the weekend". This recipe would have a weekly occurrence, it would happen once and it would last for the whole day during the weekend. An example of a rule with daily occurrence is "remind me to take the helmet." This scenario occur usually in the morning (even though it could occur more than once at different times of the day): a participant attached a token to the bicycle and another one to the helmet and implemented a rule to monitor the distance between the two. If the bicycle leaves home and the user does not have the helmet with her, she will be reminded to take it. The variety of the spectrum shows how inhabitants' needs can be quick and ephemeral or they can be more permanent. From the chart, it can be noticed at a glance that most of the envisioned scenarios consist of routines that happen daily (70%), once or repeatedly (more than three times) and at specific moments of the day (either morning, afternoon or evening). Participants, in fact, came up mainly with repeating and regular scenarios. On an axis from routine to ephemeral, most recipes involved daily and weekly habitual routines, such as "turn off the light every day when is bed-time". Some of these needs were not predictable enough to be scheduled beforehand, but they were still frequent and regular (e.g., lower heating when no one is at home, or monitor the laundry basket). Other activities, however, are occasional (14%). Participants found handful of patterns that happens occasionally and that could be implemented as recipes, such as gutter cleaning, chimney sweeping and car check-ups.

6.2.2. Trigger-action end-user programmability

After having identified the types of domestic activities and their feasibility, we explored to what extent trigger-action can express desired behaviors at home. We coded the scenarios as:

1. *Basic Trigger-Action (TA)*: the scenario can be expressed with the basic trigger-action programming approach;
2. *Advanced TA*: the scenario requires to extend the trigger-action programming approach with and/or operators (as supported by our system);
3. *Specialized Programming*: the scenario requires specialized functionality, such as a dedicated application, or it cannot be expressed as a trigger-action production rule.

The majority of envisioned scenarios belong to the *Basic TA* category (58%). The activation of the scenario can be (i) time scheduled, e.g., "if it is 6 p.m. then turn the heater on", (ii) based on sensor data, e.g., "if I'm entering home (proximity sensor), then switch on lights", or (iii) intentionally triggered by the user, e.g. "if I press a button, then send a mail". 36% of the scenarios can be expressed in terms of *Advanced TA* rules: an example is the key reminder scenario described in section 5.2.1 (see Figure 11). The remaining 6% of the scenarios are not suitable for trigger-action programming. For instance, the first family wanted a personalized shopping list application that observe food in the fridge and infer consumption behaviors in order to automatically replenish the fridge. The father of the second family wanted to record and visualize statistics about his 18 years old driving the car (e.g., speed, routes, etc.). Figure 14 shows the structure of trigger-action recipes, highlighting the combinations that required more than one trigger or action.

Figure 14: Percentages of trigger and actions combinations. Category names correspond to the number of triggers and actions. For example, (1,1) is basic trigger-action programming: 1 trigger and 1 action. 2+ means two or more triggers.

Since most scenarios (94%) can be, in principle, implemented through trigger-action programming, we analyzed combinations of triggers and actions to identify what are the technology most needed for supporting everyday routines in the domestic environment. The scenarios exploited a total of 22 different triggers that occur 152 times (Figure 15). Most common triggers are: inference of an indoor location (15.79%), time-based events (14.47%), proximity of a token from the base station or in relation with other tokens and devices (13.16%), on-demand events (11.18%), e.g., the pressure of a physical/software button, and, motion detection (8.55%). Common actions are: controlling domestic appliances (25%), receiving a notification (22.41%), modifying the normal behavior of a device or software service (16.38%), (iv) controlling lights (12.93%), and monitoring physical devices or software services (6.90%).

Triggers and actions appeared among all scenarios with 69 different combinations. Most scenarios involved home automation according to temporal events (17.40%) or proximity (8.70%), receiving notifications or reminders according to objects movement, e.g., “my father has taken his pills” (13%), or remote control of domestic appliances (8.70%).

Figure 15: Occurrences of each trigger, action, and their combinations.

6.2.3. *Integration with the landscape of existing artifacts*

The tangible approach pursued by T4Tags 2.0 —providing physical tokens that can be attached to objects— allowed participants to interact with their surrounding environment to envision and implement new activities. We therefore exploited the toolkit to examine the relationship among triggers and actions and the domestic artifacts. We categorized the home artifacts according to the Shearing Layers [Brand 1994], which organize building’s elements in the following layers (Figure 16): (i) *Site*, i.e. the geographical setting of a building, (ii) *Structure*, i.e. the foundation and load-bearing elements, (iii) *Skin*, i.e. the exterior surfaces, (iv) *Services*, i.e. the working “guts” of a building, such as communications and electrical wiring, plumbing, fire sprinkler systems, HVAC (heating, ventilating, and air conditioning), and moving parts such as elevators and escalators, (v) *Space* plan, i.e. the interior layout, such as walls, ceilings, floors and doors, and (vi) *Stuff*, i.e. chairs, desks, phones, pictures, kitchen appliances, lamps, and all the things that move around daily to monthly. Since we are interested in understanding the interactions among inhabitants and the augmented home, we extended the categorization including a further (vii) *Person* layer. Moreover, not all the interactions occur in the physical space but triggers and actions can be (viii) *Digital* (e.g., sending an email). An example of the Stuff category is the possibility of being reminded to put back the milk box in the fridge after breakfast (the scenarios was implemented by the first family, see Figure 5 and Figure 6).

Fifty-nine triggers occurrences (38.82%) are related to physical artifacts. Twenty-eight are in the *Stuff* layer and cover 19 possible different physical objects (a pack of coffee capsules, a milk box, etc.). Other 16 belong to the *Space Plan* layer and involve the placement of tokens to the floor, inside walls (e.g., sauna), the stove board, and the veranda door. For example, the first family wanted to automatically switch on the light in the bedroom when it’s dark and thus envisioned placing a token on its walls. A group of triggers occurrences (14) are in the *Skin* layer and consists on the placement of tokens both on (i) static surfaces, i.e. external walls, the lawn, or the plant soil, and (ii) dynamic surfaces, e.g., the house’s gate, main door or windows, and mail box. For instance, the third family wanted to use a token with a moisture sensor in the plant soil to sense its moisture level and switch off the watering plant system accordingly. Finally, only one trigger belongs to the *Services* layer: a token is attached to the stove hood for turning on the kitchen's ventilation system according to the steam’s volume and consistency (third family). Among triggers that are not related to physical artifacts (88, 57.89%), 57 control the location or actions performed by a person, e.g., a family member who carries a token when leaving or entering the home or who presses the button of a token, while 31 involve digital information, e.g., weather forecast data received from an Internet service.

Figure 16: Occurrences of combinations between scenarios' triggers (left side), actions (right side) and the artifacts that can be present in a home environment. The artifacts are organized according to the Shearing Layers concept [Brand 1994].

Considering the actions, a total of 101 actions occurrences (87.07%) are related to physical artifacts. Seventy-eight are in the *Stuff* layer. Some of those actions involve using the toolkit's power outlet for switching on or off domestic appliances (24), such as the Wi-Fi router, TV, car heater, stove, or the toolkits' light bulbs for controlling light settings inside or outside the house (14). Other actions require instead hacking the hardware of existing domestic appliances (4), such as the vacuum cleaner, stove's lights, microwave or dishwashing machine, or the software of smart TV, alarm clock, or drinking machine (7), for changing or augmenting their normal behavior. For example, the children of the third family wanted the smart TV to play their favorite movie when they enter a room. Other 20 actions occurrences belong to the *Services* layer. Thirteen of them involve the use of T4Tags 2.0's power plug for controlling the heating, ventilation, air conditioning, or watering plant systems. For example, the third family envisioned the use of the power plug for switching on an electric fireplace when the temperature in the living room is cold. Interestingly, the possibility to extend the toolkit's components enabled participants to envision other interactions with objects in the Stuff layer. As an example, the fourth family thought of augmenting a token with a water-level sensor and to use it with the bathtub they have in the basement (Figure 10b). They remarked that this can be useful for being notified when the bathtub is almost ready, and when it is full and thus it is time to close its tap. Participants envisioned also the attachment of a token to the crib to detect whether their youngest child woke up (with the use of a microphone), to detect whether the diapers container or the milk box are almost empty, or the waste and laundry baskets are full (with the use of a pressure sensor).

Six other actions occurrences in the *Services* layer require instead hacking the hardware of the fire alarm sprinkler system, kitchen tap, or fireplace ventilation system (4), or the software of the stove hood or the house alarm system (3). Finally, two actions occurrences belong to the *Skin* layer: one of them involve the placement of a token with a speaker to the main door to notify users when they are leaving the house without the keychain (first family), while the other requires hacking the window to automatically open it when somebody is at home in the summer (third family). Similarly, the only action that belongs to the *Space Plan* layer requires hacking the window blind in the kitchen for lowering them when it is hot and the light very bright (third family). Among the actions occurrences that do not involve any physical artifacts, 3 consist of digital actions, such as accessing an on-line resource, sending an email, or performing an on-line purchase. Others 7 regard instead visualizing information retrieved from domestic appliances, or position, movement, biometrics, magnetic, or water sensors.

6.2.4. Recipe sharing functionality

We have also investigated how to support the creation of recipes through sharing implemented recipes among families. Eleven scenarios were implemented by the first family that we made available to participants of the second and third workshops. Those families implemented 15 scenarios (8 and 7, respectively): 4 of them were adopted/tailored from existing recipes in the public library. As an example, the second family found useful the public recipe "Keys reminder" (Figure 11): they adopted and personalized the behavior by choosing the message for the notification. The third family, instead, used a public recipe as a source of inspiration. While browsing the list and thinking of the needs they might have, they saw the "Sauna Ready" recipe —checking when the sauna is ready with the temperature sensor and send a message— and said:

– Mother: "*Here, 'sauna is ready'";*
– Father: "*So why we don't do this? when you have fever, we'll send the email to you [referring to his wife]*";
– Mother: "*Yes that could be one! A child puts this one on her head [she performed the gesture of putting a token on the forehead], it sends mother and father a message that the…*";
– Father: *"No, to the teacher directly";*
– Mother: *"Yeah! [enthusiastic and laughing] […]".*

To this end, they created a recipe for sending an email to the kid's school when her temperature goes above 37°, and then acted out such situation (see Figure 17). Of course, the token technology is not reliable for implementing this scenario. In any case, this process shows how the recipe sharing functionality can be used to inspire new appropriations.

Figure 17: (a) Using of a token to sense the child's temperature. (b) Simulating fever by attaching the token to the kettle. (c) Sending an email to the child's school for notifying her absence.

6.2.5. *Enabling children as design participants*

Our approach to design domestic technologies in-situ with and by the user allowed to explore scenarios that emerged from children direct experience. Even if programming recipes using T4Tags 2.0 was too difficult for children, children engaged with the physical tokens with sustained interest and curiosity, providing ideas and acting out different scenarios, such as augmenting posters or toys with multimedia content. The physical affordances of the tokens (e.g., tokens can be easily attached to objects, they have different colors, they provide physical buttons, etc...) allowed children to experiment with the system in the domestic environment and, as a result, researchers were able to involve children as design participants and gather information that was not filtered by adults' interpretation of children's experiences. Technology immersion allowed to overcome the language barrier during the brainstorming session. Younger children (age 7 or less), in fact, could not speak English and researchers, who were not Finnish speaking, had to rely on adults' translation from Finnish to involve children in the ideation process. However, observing children interacting with the system could overcome this barrier, and it provided valuable design material to address children perspectives on matters that affect their lives.

7. DISCUSSION OF FINDINGS

In addition to descriptive data, themes emerged that are critical for the design of domestic technologies that can easily reconfigured by the users.

Trigger-action programming for the home: Independently of the availability of interconnected smart appliances, an approach to trigger-action programming which allows to combine multiple triggers, demonstrated to be able to provide enough expressiveness to describe most of the desired behaviors to meet inhabitants' needs (a total of 94% of scenarios). This is resonant with findings from Ur et al. [2014]. Their investigation showed that the majority of smart home behaviors submitted by Amazon Mechanical Turk participants centered on trigger-action programming. Our results confirm their findings in a more ecologically valid setting and show that trigger-action is a practical approach for end users. It provides the right level of abstraction to program the behavior of smart devices (e.g., power plugs or light bulbs) as well as domestic objects augmented with sensing technology. However, participants were not able to implement some of such scenarios that could be expressed as trigger-action productions, either because the toolkit did not provide the appropriate technology or, more interesting, because they were not able to express the behavior as composition of triggers and actions. For instance, in the case of the "key reminder" scenario in Figure 11, participants were not able to define the more complex situation that takes into account also who is the owner of the keys. Indeed, extending simple trigger-action programming with and/or composition enabled many scenarios that could not have been possible to program otherwise. We, therefore, confirm that it is a promising approach for programming the domestic environment [Ur et al. 2014]. Even so, the interaction with physical objects showed that a wide variety of complex situations are difficult for the users to express as trigger-action combinations. Participants were able to devise needs in terms of an "if-this-then" structure, but their implementation was not always straightforward. Take as an example the "if the fridge door is open for long, then beep an alert" behavior. It was not straightforward for the participants to find a composition of triggers to program the "open for long" behavior. Our findings suggest that allowing users to augment physical objects at home increases the type of activities that can be supported, but at the same time, increases the complexity of the trigger-action paradigm. It is paramount to further investigate how to enable end users in expressing desired behaviors, especially when interacting with physical objects.

In pursuing end-user programming of the domestic environment, different levels of programmability for supporting the creation of complex behaviors should be considered. Recent work has explored different alternatives to describe and program trigger-action rules, which include the use of WHEN and WHILE instructions, in an attempt to increase the expressivity of the rule-based language and make it more "natural" for the end user [Ghiani et al. 2017]. The in situ experimentation with our programming toolkit confirmed the need for seeking alternative programming structures. For example, linking together more recipes with a "if recipe1 AND if recipe2 AND if recipe3…" or "if recipe1 ELSE IF recipe2 ELSE IF …" structure can allow to program behaviors that are based on different parallel or concurrent conditions, respectively. One example in which those programming structures would be useful is given by the scenario depicted in Figure 10, where users needed to create two different rules to turn on and off the veranda light when the veranda door was opened and closed. However, augmenting the expressiveness of the language to program recipes also introduces issues for the accuracy of people's mental model [Huang and Cakmak 2015]. It is important, for instance, to understand possible causes of ambiguities in both interpreting and creating rules that could lead to unwanted behaviors. A crowd-based library of implemented recipes could improve the accessibility of end-user programming by fostering self-reflection, learn-by-doing and support problem-solving in the existing end-user programming environment, encouraging users to learn from others.

During the workshops, participants felt the need for implicitly reverting an actuator to its original state when defining rules with actions that terminate after a certain amount of time (extended actions according to the definition from Huang and Cakmak [2015]). Since the current version of the toolkit does not support the implicit creation of opposite rules, participants had to explicitly create the rule by themselves, but reported that an automatic behavior would benefit the programmability. For instance, the first three families expressed the need to always have the WiFi Internet connection on, and to switch it off only at specific times of the day, e.g., at night. The fourth family desired to automatically switch off some appliances when they were leaving the house and switch them on again when returning back home.

Conversely from other studies that used a pen-and-paper approach [Brich et al. 2017] or exploit a digital tool like IFTTT [Ur et al. 2015] to elicit possible scenarios that could be implemented with trigger-action programming at home, our tangible toolkit invited users to physically interact with things at home and explore objects that could be augmented with technology, thus discovering new opportunities for triggers and actions. Tangible interaction showed that attaching sensors to something physical could be sometimes an obstacle to interaction; for some scenarios the token metaphor is not the most effective, for instance, monitor if the milk is outside the fridge could be implemented through proximity sensors or load cells. In other cases, tokens provided new affordances and helpful handles of information that eventually would enhance the user experience with the smart home (e.g., use a physical token to share a to-do list or the "play the guitar" scenario in Figure 18).

Sharing implemented usages for inspiring users: Community creation of a library of shared recipes provides domestic end-user programming with a means to foster and sustain user engagement and creative ideation. Recipe sharing enables: (i) the capitalization of the expertise of other users and be inspired by existing examples of system use, (ii) the customization of existing examples and create new experiences, and (iii) receiving guidance in putting into practice the technical assembly of the hardware and software components. If a tech-savvy user creates a technological implementation for a common domestic problem, this can be adopted by non-technical users without having to struggle to find the right composition of services and sensors. Crowd-fueled implementations takes end-user programming for the home one step further. If, by creating a recipe, a user creates a completely new functionality that is useful for his or her family, other users can adopt it and, thus, increase the utility and value of artifacts and devices that are already in their homes.

In this paper, we detailed how we found a great degree of diversity in the kind of activities that can be programmed in the domestic environment. Therefore, concerning the scalability of the recipes sharing approach, as more recipes are shared, the users could find it difficult to find useful recipes in the library. Different browsing, filtering and recommending approaches need to be implemented to help users navigate through recipes, as well as metrics to evaluate when the system effectively match with a user need. Indeed, browsing a collection of implemented recipes can help families find very common, or very complex household behaviors and support users thinking creatively, iterating and adapting. For instance, one user could learn that a temperature sensor is a feasible technology to check whether the milk has been outside the fridge for too long.

A major shortcoming observed during the workshops is that users were not keen to share their recipes with others. As explicitly remarked by participants, they perceived sharing information as a privacy threat. During the workshops, in fact, it was noted that such participants preferred to set the recipes as private, that is, they were concerned that sharing recipes would constitute a privacy threat if it would reveal too much sensitive information, e.g., information that would allow to identify the family members and their routine. Participants reported that the user interface does not clearly indicate what are the information that will be shared with other users, and they would not want to reveal the specific details of their everyday routines. Sharing recipes anonymously was proposed as a possible solution to the privacy issue.

Users' concern about privacy and security issues with ubiquitous technologies are not new a topic in the research literature. Dourish et al. [2004] reflected on the need to manage privacy as a mean to increase security and support the adoption of ubiquitous technologies. As we have experienced in our experimentation, privacy management has a central role for the future adoption of recipe sharing mechanisms at home. Even if participants acknowledge the benefits of having a shared database of ideas, they would be reluctant to use it unless the system is able to provide a desired level of privacy and security, and the user interface is designed to appropriately convey that information. Our in situ investigation also confirms qualitative findings from other studies on privacy for domestic systems that were conducted either through phone interviews [Zeng et al. 2017] or in laboratory setting [Carruccio et al. 2015]. Privacy concerns deter user from engaging with a smart home system and the user interface of such systems should be purposefully designed to provide awareness and control over privacy settings.

Integration with home technology: The complexity of the domestic environment [Crabtree and Tolmie 2016] calls for an action to radically re-think end-user systems. They should not only provide capabilities to program and configure a single device, but instead they should enable user to integrate technology in the often too cluttered and ill-defined landscape of artifacts and domestic activities. Users demand for better "control over their lives" [Davidoff et al. 2006] by seamlessly and quickly adapting technology to their needs. This requires understanding the role that the technology plays in augmenting the existing artifacts and appliances at home, which in turn urges the importance of shifting from a technology-driven to a user-driven perspective. To be "smart", in the case of end-users' domestic technologies, means to empower users to exploit and re-configure the domestic space in a meaningful way.

Compared with other studies on trigger-action programming in the smart home [Huang and Cakmak 2015; Ur et al. 2014], the in-situ experimentation uncovered a wide variety of scenarios that involved articulated and unexpected users' interactions with the different physical layers of the home to support their activities, in addition to simple home automation behaviors [Brich et al. 2017]. Our design activities show that potentially people would like to integrate much more the technology with the domestic environment but they are currently constrained by the lack of technological substratum. The integration with domestic appliances, for instance, was not yet feasible because it required hacking the capabilities and required for the appliances to expose their functionality as web-enabled services that could then be exploited by the toolkit. This reveals many design challenges that are the focus of investigation for the technical feasibility of the smart home, such as device interoperability, cross-device functionality, and home operating systems that are able to manage the ecosystem of smart devices [Dixon et al. 2010].

While a trigger-action approach with multiple triggers is effective for programming most scenarios, our results show that the tangible manipulation of augmented artifacts provides an extra degree of programmability to define smart behaviors of objects. Attaching tokens to objects enabled "real-world tangibility" [Lee et al. 2013], that is, users interacted with real world objects and this made it easier to implement desired behaviors and also to unveil new potential activities and technological appropriations. Tangible augmentation proved to be an important enabler for people to meaningfully interact with their home, by manipulating and using the things they have around and re-organize them in creative ways. We have, however, found limitations in the current token technology. One is the lack of potentially useful sensors, which could be easily offset with the integration of new specialized modules. Additionally, the token size and shape, hinder their integration with small and movable objects. Further investigation is needed to explore how to design tokens with different materials and shapes. For example, thinner tokens with only a proximity sensor can be attached to the back of remote controllers or wallets and used to track their presence inside or outside the home. Flexible tokens can afford to be shaped in different ways, such as a collar, and used for monitoring pets, or placed around the pencil case or bottles.

The integration with the domestic environment also unveils the need for supporting open-endedness and repurposability of the technology. Many of the scenarios implemented during the workshops, in fact, indicate how dedicated devices are not a sustainable choice. Dedicated devices, in fact, are discarded as soon as their service is no longer required, because they cannot be recomposed or adapted to the needs of new situations. T4Tags 2.0, instead, allows repurposability because it was designed as an open-ended technology to be completed by users according to their situated experiences and needs. For instance, the second family envisioned to use a token in summer to sense the brightness of light and lower the windows blinds accordingly (together with an actuator). A token can also be used for making a power plug switch on a winter lamp in the morning when it is dark outside and people are entering the kitchen. Both scenarios can be supported by the same token that can be repurposed at the end of each summer or winter.

Overall, while on one hand the current shape and size of tokens allowed users to extend most of their home artifacts, on the other hand, results suggest that other shape and materials should be explored for enabling tokens to be more integrated with people's everyday routines that involve movable objects. In particular, following Taylor and Swan [2005] design implications for home information devices, designers of end-user technologies should consider the purpose and the way of usage of portable artifacts. Then, they should carefully select and embed into tokens only those sensors needed to support users' desired usages with such artefacts. For example, thinner tokens with only a proximity sensor can be attached to the back of smartphones or remote controllers and be used to track their presence inside or outside the home. Similarly, smaller tokens with a more prominent round hole on the top can be attached to the key chain, while flexible tokens can afford to be shaped in different ways, such as a collar and be used for monitoring pets, or placed around the pencil case. Flat tokens with only a motion sensor can instead afford to be attached to the top of the pill's box or the back of fridge products to track and notify when some movements have been performed with them.

As emerged from the workshops, more flexible tokens might also afford to be used in other contexts outside the home environment. Interestingly, this emerged from the family member, i.e. the mother, who showed a conservative attitude toward such kind of technology. Indeed, although she said that its use can be detrimental for people, since it can make them dependent on it and decrease their memory abilities, in line with what emerged from the study by Mäyrä at et al. [2006]. The mother outlined that this technology can be very useful for people with disabilities. For example, she envisioned the attachment of small tokens to toys for creating a reward system for children with learning disabilities in structured learning task, e.g. a tablet app asks a child to bring back the red ball and rewards him accordingly.

Using the environment as interface for programming: Many scenarios show that simple motion sensing capabilities are not enough to enable smart behaviors since users expressed the need to be able to define custom gestures and movements when manipulating physical artifacts. An approach to end-user programming that capitalizes on programming-by-demonstration was identified as a possible solution to this problem during the workshops. Programming-by-demonstration [Lieberman 2001] would allow the users to program smart behaviors by using the physical environment as interface for programming. Instead of defining numerical values that are difficult to associate to their physical counterpart, or interpreting vague labels, e.g., the light is "dim", users can program a behavior by letting the system inferring sensor data from an actual real-world situation. For instance, users can show the system what do they mean with "dim" by programming the light sensor trigger with real-time data from the light sensor itself. Programming the system by showing examples of interactions would also allow users to increase the capability of the system with user-generated triggers, that would be beneficial also in terms of sharing solutions with others.

Figure 18: Participants acting out a scenario for checking if the daughter is playing the guitar.

Some scenarios involved the use of custom manipulations to activate the trigger. As an example, the parents of the first family in the elicitation study wanted to ensure that their daughter is keeping up with the practice of a music instrument. They envisioned a recipe for detecting when the daughter starts/finishes practicing with the guitar. The parents reported that a token could be attached behind the headstock (see Figure 18) and they then envisioned that the system could be trained to recognize three gestures: one when the daughter picks up the music instrument from the stand, another one when she puts it down, and a third one that would monitor how long the daughter is playing by recognizing when the daughter is holding the guitar. The scenario was only acted out by the family during the workshop because it would have had required the implementation of a custom gesture. In this case the precision and recall of the gestures and manipulations a of users are crucial. To this end, it would be important to understand how to design pattern matching systems with graphical interfaces to support editing and recognition of matching criteria. For instance, if a wide tolerance is required to make the system work, how would the system handle the interference between multiple gestures that have been defined? Would gestures essentially have to be extremely distinct, and what would that imply about the scalability of the system? Given the aforementioned issues, while it would not be difficult to implement programming-by-demonstration for one-dimensional data like proximity, light intensity or humidity, developing a recognition engine for custom gestures and object manipulations would introduce a big implementation burden.

8. LIMITATIONS

In this section, we analyze some factors that could have limited the validity, reliability, and generalizability of our study and the actions taken to overcome them. Since the study described in this paper is qualitative in nature, we followed the terminology defined by Noble and Smith [2015] for evaluating quality in qualitative research.

- *Truth value (or credibility)*: Since the conclusions of the study are based on researchers' interpretations of participants' activities and reflections during the workshops, a possible limitation regards the fact that researchers' personal experiences and viewpoints might have biased the results. Although a small degree of interpretation might be unavoidable in qualitative studies, to enhance truth value in this investigation we documented all workshops with video recordings and field notes, to be able to review them at any point and verify our findings. Moreover, we included in the paper several participants' observations and quotes from the workshops to highlight their perspectives and support our results.
- *Consistency/Neutrality*: Although each workshop aimed at investigating different aspects related to the design of domestic end-user systems, to ensure consistency and neutrality, we adopted the same protocol for each study session and the same methodologies for documenting them, which we described in the Procedure section. However, a possible limitation of our findings consists in the fact that all participants were exposed to a specific toolkit which could not support all the

scenarios needed by them. Moreover, the first family couldn't access the Recipe Library. This could have resulted in the lack of probes for envisioning possible usages, thus hindering participants' creativity. Future studies should then consider using more and different technologies and pre-made scenarios for the home environment to explore a wider range of users' needs.

- *Applicability (or transferability)*: A possible limitation of our study consists in the fact that the families were all Finnish, and results might not be easily generalized to other contexts. For example, some of participants' scenarios were closely linked to the local habits and weather conditions, e.g. those related to the sauna usage or home- and car-heating needs during the winter. Thus, future investigations should consider participants from different countries. Future studies should also include participants with experience in smart-home technology to have insights on their everyday engagement with such systems.

Issues with conflicting goals were raised during workshops, for example a window that should be open or close according to different goals such as safety or comfort. Participants felt the need to have a veto power to stop or revert the effect of automatic rules, confirming the results from previous works [Berch et al. 2017]. Related to this issue, participants also expressed the need to have an interface, possibly an app for their mobile phone, that allows them to be aware of the current state of the system at any time. In this way it would be easier for them to stop or revert a rule that is currently active. Even if this aspect is quite relevant for the development of domestic technology, we did not analyze it in depth in the current study and left the investigation open for future work.

Lastly, the study focused on understanding how to integrate technology in the domestic environment and artifacts and to what extent trigger-action programming is a good enough model to support end-user interactions with existing objects. We did not evaluate the usability of the proposed tool, for example we did not evaluate how the proposed user interface metaphor supports or hinders the creation of complex trigger-action compositions when compared to existing tools from the state of the art. We observed that sharing recipes allows non-technical users to build on the experience of other users. However, our study was limited in time and therefore we were not able to study the magnitude of this effect, that is, whether users only adopt shared recipes for a particular need or they effectively learn from others how to express desired behaviors as composition of given sensors and actuators, which ultimately would change their mental model for trigger-action programming.

9. CONCLUSIONS

In the domestic setting, end-user programming is a more complex activity than merely *configuring* things: the inherent complexity of the home brings people towards the need of *programming* smart behaviors of interconnected things. In such an ill-defined scenario, an approach to trigger-action programming with multiple triggers demonstrated to be practical to express the bulk of desired behaviors. However, more advanced programming spurs novel uses. We have shown, in fact, that pursuing tangible interaction, e.g., tokens that can be attached to objects and augment their capabilities, enabled users in implementing meaningful interactions and uncovered the need of using the physical environment as interface for programming through programming-by-demonstration approaches.

We remark that our design journey highlighted different avenues for further investigation. First, we confirmed the results of other studies conducted in artificial settings and showed that trigger-action programming with multiple triggers could lead to ambiguities both in the composition of triggers and actions as well as in the interpretation of the outcome. Therefore, additional field studies are required to understand the users' mental model and how they can be empowered to express their needs as trigger-action rules. Second, there is still a lot that we can learn *from the wild* and longitudinal studies are needed to better understand long-term usage once the system's novelty wears off and gain insights that are more representative of the normal use of the technology. Lastly, we plan to adopt a "research in the large" [Kranz et al. 2013] approach and ship the toolkit to several families. This way, a critical mass of users would be reached easier and the implications of the recipe sharing platform would be studied in a more ecologically valid setting. Crowd-based sharing of implemented recipes showed potential to be a desirable platform for harnessing users creative input, fostering sustained engagement and helping users helping themselves, by learning from other people's examples, with some privacy concerns.

10. ACKNOWLEDGEMENTS

This work was supported by the 2015 UC3M Mobility Grant and by the Academy of Finland (286440, Evidence)

REFERENCES

Karuppanan Balasubramanian and Akin Cellatoglu. 2008. Improvements in home automation strategies for designing apparatus for efficient smart home. *Consumer Electronics, IEEE Transactions on.* 54, 4 (2008), 1681–1687.

Andrea Bellucci, et al. 2014. Prototyping device ecologies: Physical to digital and viceversa. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction (TEI '14).* ACM Press, 373–376.

Andrea Bellucci, et al. 2015. Extreme co-design: Prototyping with and by the user for appropriation of web-connected tags. In *Proceedings of the Fifth International Symposium on End-User Development (IS-EUD '15).* Springer International Publishing, 109–124.

Bellucci, Andrea, et al. 2016. Supporting the serendipitous use of domestic technologies. *IEEE Pervasive Computing*, 15, 2 (2016), 16-25.

Hugh Beyer and Karen Holtzblatt. 1997. Contextual design: defining customer-centered systems. Elsevier.

Alan F. Blackwell and Rob Hague. 2001. AutoHAN: An architecture for programming the home. In *Human-Centric Computing Languages and Environments, 2001. Proceedings IEEE Symposia on*. IEEE, 150–157.

Stewart Brand. 1994. How buildings learn: What happens after they're built. New York: Viking Press.

Jill C. Cao. 2013. *Helping end-user programmers help themselves -- the Idea Garden approach*. Ph.D. Dissertation. Oregon State University, Corvallis, OR, USA.

Marshini Chetty, David Tran, and Rebecca E. Grinter. 2008. Getting to green: Understanding resource consumption in the home. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp '08).* ACM Press, 242–251.

Jeannette S. Chin, Vic Callaghan, and Graham Clarke. 2006. An end-user programming paradigm for pervasive computing applications. In *Pervasive Services, 2006 ACS/IEEE International Conference on.* IEEE, 325–328.

Diane J. Cook, et al. 2013. CASAS: A Smart Home in a Box. *Computer*. 46, 7. 10.1109/MC.2012.328.

Andy Crabtree and Peter Tolmie. 2016. A day in the life of things in the home. Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing. ACM, 2016.

Scott Davidoff, Min Kyung Lee, Charles Yiu, John Zimmerman, and Anind K. Dey. 2006. Principles of smart home control. In *UbiComp 2006: Ubiquitous Computing*. Springer Berlin Heidelberg, 19–34.

Anind K. Dey, et al. 2006. iCAP: Interactive prototyping of context-aware applications. *Lect. Notes Comput. Sci*. 3968 (2006), 254–271.

Audrey Desjardins and Ron Wakkary. 2013. Manifestations of everyday design: guiding goals and motivations. In *Proceedings of the 9th ACM Conference on Creativity & Cognition.* ACM Press, 253–262.

Luigi De Russis and Fulvio Corno. 2015. HomeRules: A tangible end-user programming interface for smart homes. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15).* ACM Press, 2109–2114.

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, Article 12 (April 2017)*, 52 pages.

Colin Dixon et al. 2010. The home needs an operating system (and an app store). In *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets '10).* ACM Press, 1–6.

Paul Dourish, Rebecca E. Grinter, Jessica Delgado de la Flor, and Melissa Joseph. 2004. Security in the wild: user strategies for managing security as an everyday, practical problem. *Personal and Ubiquitous Computing*, 8(6), 391-401.

David Frohlich and Robert Kraut. 2003. The social context of home computing. In *Inside the Smart Home.* Springer London, 127–162.

Sumit Gulwani, Jose Hernández-Orallo, Emanuel Kitzelmann, Stephen H. Muggleton, Ute Schmid, and Benjamin Zorn. 2015. Inductive Programming Meets the Real World. *Commun. ACM* 58, 11 (2015), 90–99.

Pam Heath and Nancy Bell. 2006. The changing world of home technology: A Microsoft case study. *Inf. Soc.* 22, 4 (2006), 251–259.

James Hollan, Edwin Hutchins, and David Kirsh. 2000. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7, 2 (2000), 174-196.

Jiri Hosek, et al. 2014. Universal smart energy communication platform. In *Intelligent Green Building and Smart Grid (IGBSG), 2014 International Conference on.* IEEE. 1–4.

Steve Howard, Jesper Kjeldskov, and Mikael B. Skov. 2007. Pervasive computing in the domestic space. *Pers. Ubiquitous Comput.* 11, 5 (2007), 329–333.

Justin Huang and Maya Cakmak. 2015. Supporting mental model accuracy in trigger-action programming. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15).* ACM Press, 215–225.

Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. 2003. Technology probes: inspiring design for and with families. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03). ACM, New York, NY, USA, 17-24.

Amy S. Hwang and Jesse Hoey. 2012. Smart home, the next generation: Closing the gap between users and technology. In *AAAI Fall Symposium: Artificial Intelligence for Gerontechnology*.

Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '97).* ACM Press, 234–241.

Timo Jakobi, Corinna Ogonowski, Nico Castelli, Gunnar Stevens, and Volker Wulf. 2017. The Catch(es) with Smart Home: Experiences of a Living Lab Field Study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17).* ACM, 1620–1633.

Tejinder K. Judge and Carman Neustaedter. 2014. An introduction to studying and designing technology for domestic life. In *Studying and designing technology for domestic life: Lessons from home*. Elsevier, 1–8.

David Kirsh. 2017. Embodied cognition and the magical future of interaction design. *ACM Trans. Comput.-Hum. Interact.* 20(1), Article 3 (2013), 30 pages.

Matthias Kranz, Lukas Murmann, and Florian Michahelles. 2013. Research in the large: Challenges for large-scale mobile application research-a case study about NFC adoption using gamification via an app store. *International Journal of Mobile Human Computer Interaction (IJMHCI),* 5, 1, 45-61.

Jisoo Lee, et al. 2013. A tangible programming tool for creation of context-aware applications. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13).* ACM Press, 391–400.

Jung-Joo Lee, et al. 2014. Domestic appropriations of tokens to the web. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14).* ACM Press, 53–62.

Henry Lieberman. 2001. Your wish is my command: Programming by example. Morgan Kaufmann.

Youn-kyung Lim, et al. 2013. Discovery-driven prototyping. *Pervasive Comput. IEEE* 12, 3 (2013), 74–80.

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. *Lect. Notes Comput. Sci.* 7319 (2012), 143–160.

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 *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14).* ACM Press, 105–115.

Frans Mäyrä, et al. 2006. Probing a proactive home: Challenges in researching and designing everyday smart environments. *Hum. Technol.* 2, 2 (2006), 158–186.

Mark W. Newman. 2006. Now we're cooking: Recipes for end-user service composition in the digital home. In *CHI 2006 Workshop IT@Home*.

Seymour Papert. Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc., 1980.

Neal Rosen, et al. 2004. HomeOS: Context-aware home connectivity. In *International Conference on Wireless Networks.* 739–744.

Corina Sas and Carman Neustaedter. Exploring DIY practices of complex home technologies. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2017, 24, 2 (2017), 16.

Albrecht Schmidt, Lucia Terrenghi, and Paul Holleis. 2007. Methods and guidelines for the design and development of domestic ubiquitous computing applications. *Pers. Ubiquitous Comput.* 3, 6 (2007), 721–738.

Alex S. Taylor and Laurel Swan. 2005. Artful systems in the home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05*). ACM Press, 641–650.

Turkle, Sherry. "How computers change the way we think." Law and Society Approaches to Cyberspace. Routledge, 2017. 3-7.

Blase Ur, et al. 2014. Practical trigger-action programming in the smart home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14).* ACM Press, 803–812.

Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, Jiyun Lee, Sarah Mennicken, Noah Picard, Diane Schulze, and Michael L. Littman. 2016. 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 (CHI '16).* ACM Press, 3227–3231.

Tim L. M. van Kasteren, Gwenn Englebienne, and Ben J. A. Kröse. 2011. Hierarchical activity recognition using automatically clustered actions. In *Proceedings of the Second International Joint Conference on Ambient Intelligence*. Springer Berlin Heidelberg. 82–91.

Frank Vetere and Frank Feltham. 2007. Bringing emotion and physicality to domestic ICTs: interview with Steven Kyffin. *Pers. Ubiquitous Comput.* 11, 5 (2017), 335–337.

Jong-bum Woo and Youn-kyung Lim. 2015. User experience in do-it-yourself-style smart homes. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15).* ACM Press, 779–790.

Charlie Wilson, Tom Hargreaves, and Richard Hauxwell-Baldwin. 2015. Smart homes and their users: A systematic analysis and key challenges. *Pers. Ubiquitous Comput.* 19, 2 (2015), 463–476.

Salu Ylirisku et al. 2013. Designing web-connected physical artefacts for the "aesthetic" of the home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13).* ACM Press, 909–918.

John Zimmerman, Jodi Forlizzi, and Shelley Evenson. 2007. Research through design as a method for interaction design research in HCI. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 493-502.

LIST OF FIGURES

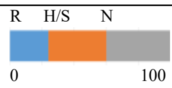
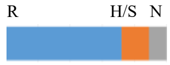
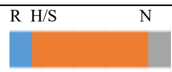
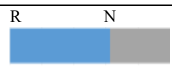
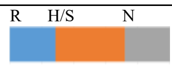
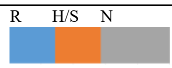
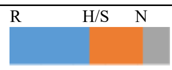
Table

Table 1. Details of the families involved in the three workshops.

Family	Demographics	Living situation and technology use
F1	Both the father (37) and the mother (37) works in the academia: they are researchers in electrical engineering and industrial design. They have two daughters (7 and 4).	They live in a spacious flat. They own a desktop computer and two tablets that are used for work (father) or leisure time (mother and daughters).
F2	The father (41) is an interaction designer, while the mother is a housewife (38). They have two sons (9 and 7).	They live in a row house. They own two tablets, two TVs (one in the basement for the children), and a desktop computer.
F3	The father (39) is an engineer, while the mother (38) works in an open technology innovation organization where she is doing also a PhD. They have two daughters (9 and 7).	They live in a spacious detached house. The father likes to manage the technology networks and implementations in the house, while the mother is enthusiastic about automation gadgets and applications. The children have their own "disco" lighting in the basement next to a home cinema set with a projector.
F4	The father (37) has a major in philosophy and is currently attending a high school teaching preparation course, the mother (37) works in a kindergarten. They have three sons (16, 15 and 2).	They live in a detached house organized in two stories and a basement, with an independent garage. At home they have four computers (one each except for Sam), and one tablet that they use with the Chrome Cast to see Netflix on their TV. The two older sons have a PlayStation 4 in their room.

Table

Table 2. Categorization of scenarios according to their high-level goal[1].

High-level goal	Description	Frequency	Feasibility
Comfort	Tasks for increasing users' comfort, e.g. automatically turn on the airflow control when it's hot (I F3), or preventing disturbing or annoying situations, e.g., turn off the fire alarms' blinking lights during night.	20%	
Safety	Tasks for guaranteeing users' safety, e.g. switch off the stove to prevent fire (E F1, F2).	17%	
Parenting	Tasks related to caring of both children and grandparents, e.g. send a notification when a child comes home (I F1), or set a timer to switch off children's on-line devices (E F1, F2; I F3).	12%	
Resource Conservation	Tasks to conserve natural resources like water and/or energy, e.g. switch off the washing machine when it has finished (E F2).	9%	
Health & Fitness	Tasks for preserving and/or increasing people's health, or monitoring physical exercise, e.g. send a notification when grandparents took the pill (I F1).	8%	
Home Maintenance	Tasks related to the maintenance of the house and its components, e.g. watering plants (E F1, F3), or automatic cleaning (E F1).	8%	
Object Reminder	Tasks for avoiding forgetting objects, e.g. send a notification to remember to pick up the soccer gear when leaving the home, or to get the keys to avoid remaining outside in the cold (I F1).	7%	
Entertainment	Tasks related to entertainment, e.g. play music when somebody enters the home (I F3).	5%	
Security	Tasks for guaranteeing people's security e.g., trigger burglars' alarm (E F2).	5%	
Cooking	Tasks related to cooking, e.g., switch on the oven when a person is on her way home (E F2).	5%	
Education	Tasks related to education, e.g., check if a child is practicing with a music instrument (I F3)	4%	

1 The table reports the frequency of scenarios per category and the overall feasibility of each category. Examples of scenarios that have been enacted (E) or implemented (I) by the three families are provided. In the Feasibility column, the visualization shows the percentage of scenarios for each category that: are readily implementable (R), need minor hardware/software adaptations (H/S) or, cannot be implemented (N).

Figure

Figure 1: The T4Tags 2.0 toolkit.

Figure

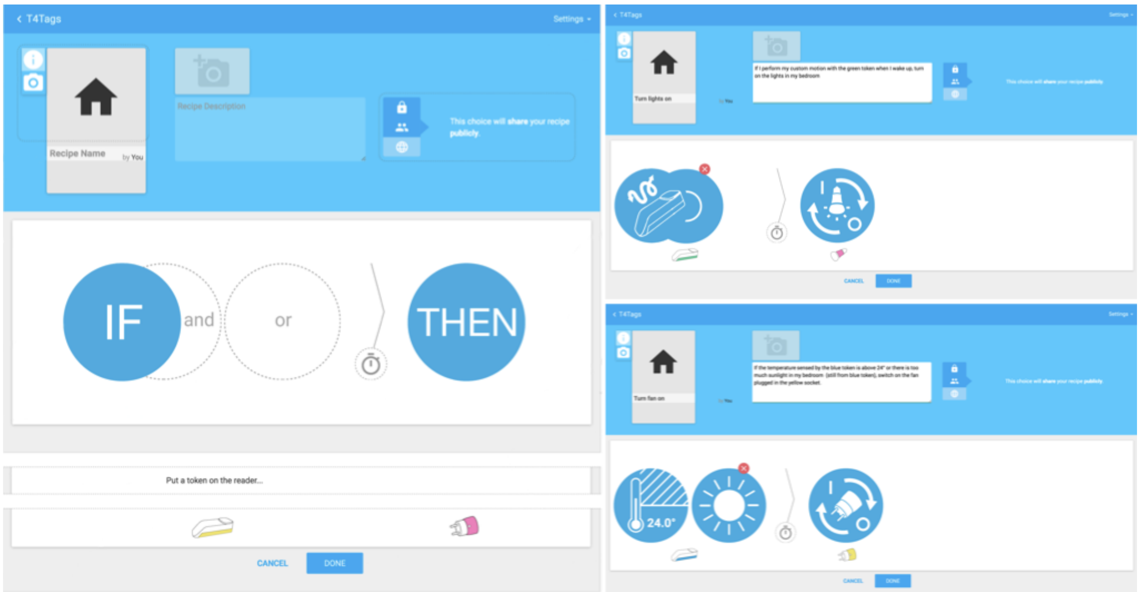


Figure 2: The user interface of T4Tags 2.0 to create "recipes" for the tokens. On the left the main interface and on the right two examples of AND/OR compositions.

Figure

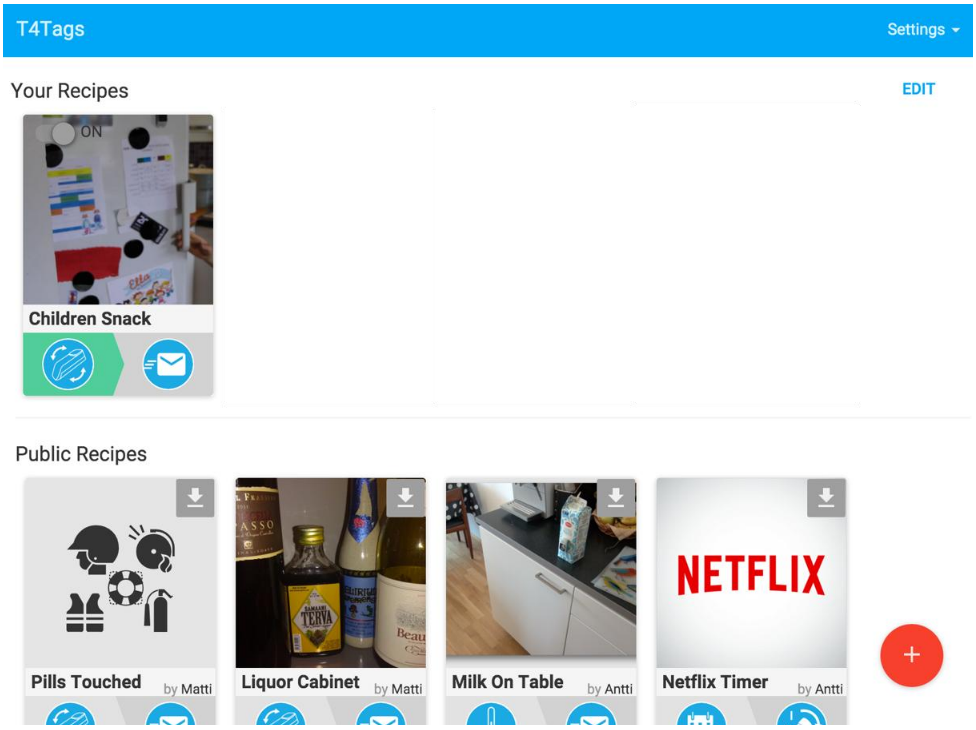


Figure 3: The “Recipe browser” page that shows the recipes created by the user (“Your Recipes” section), and those created and shared by others (“Public Recipes” section).

Figure

Figure 4: A sample scenario for the kitchen. (a) A token is attached to the milk box. (b) The token senses the milk box temperature when left on the kitchen counter and blinks or play a sound when it reaches 20 degrees. (c) Users are reminded to put the milk box back in the fridge.

Figure

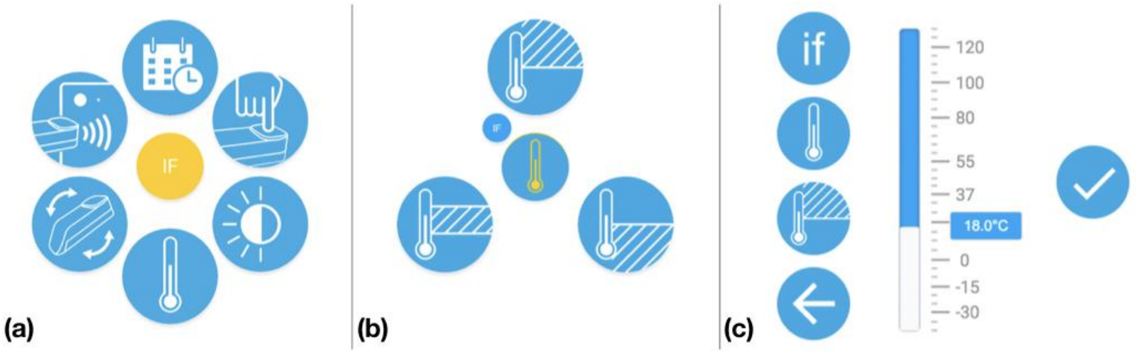


Figure 5: The user interface to select triggers. (a) Available triggers. (b) Available options for the trigger "temperature." (c) User interface for the definition of the temperature.

Figure

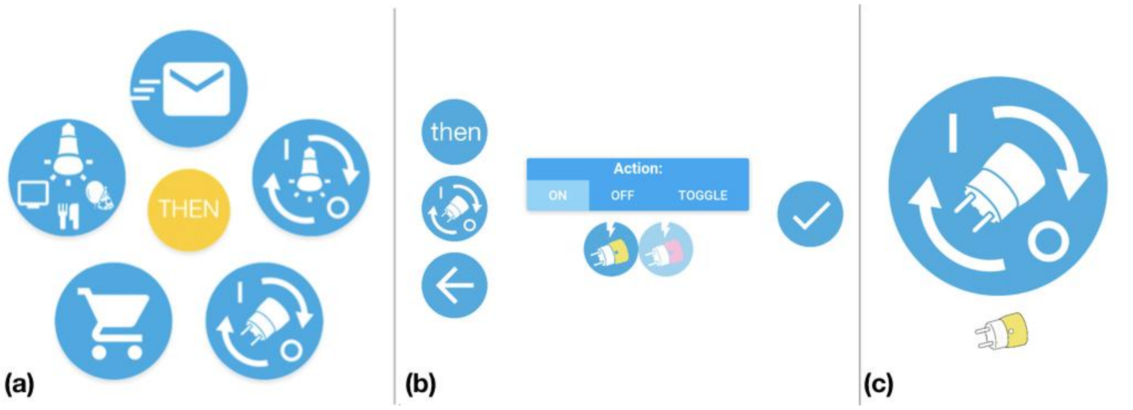


Figure 6: The user interface to select actions. (a) Available actions. (b) Available options for the action "play sound." (c) The system invites the user to place a token on the wooden case to assign the recipe to it.

Figure

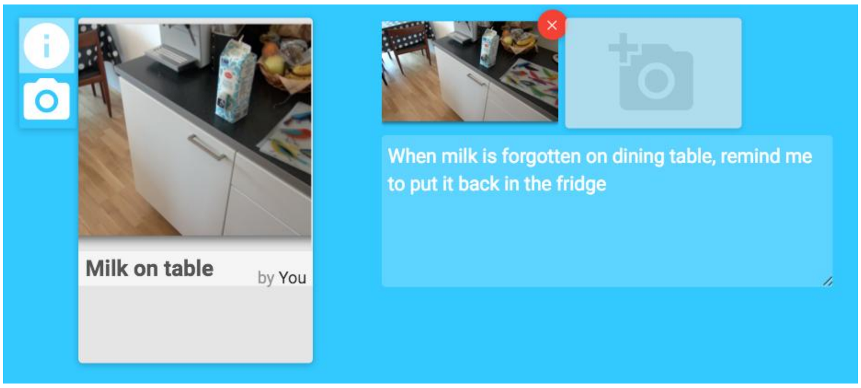


Figure 7: Picture uploaded by the mother to better describe the behavior of the recipe.

Figure

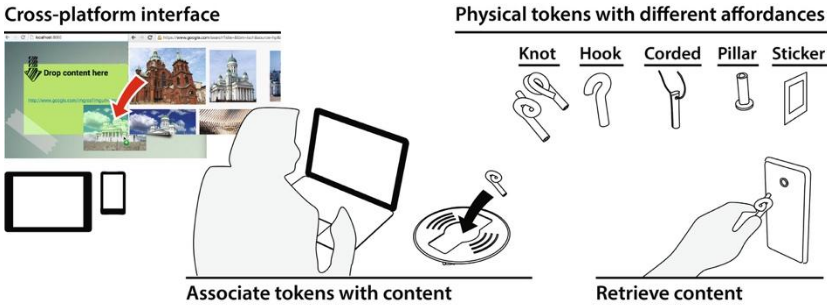


Figure 8: The T4Tags 1.0 open toolkit.

Figure

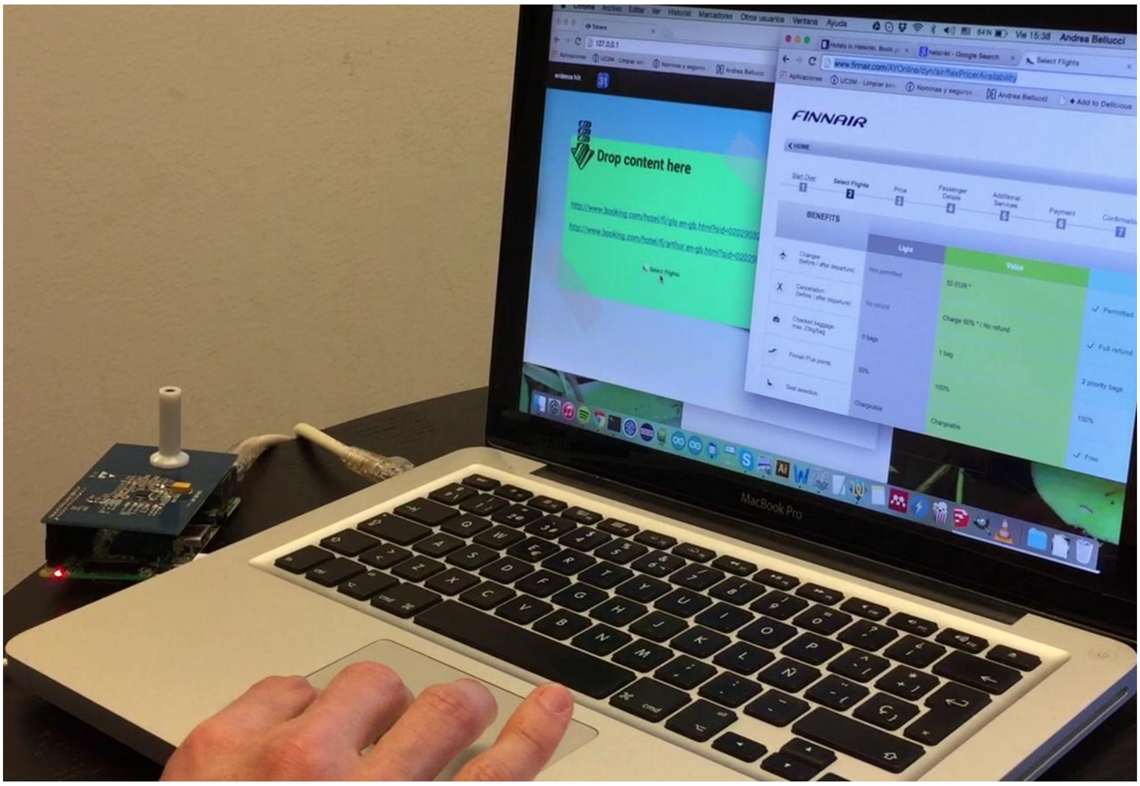


Figure 9: The implementation of a scenario. A token is used to store information to plan a family trip: hotel booking, flights and images of different locations to visit.

Figure

Figure 10: (a) Paper-based mockups of sensors and actuators that were provided to participants. They were also given blank mockups on which they could sketch desired technology. (b) Participants extended a token with a "water-level" sensor (they used the available mockup for a moisture sensor) and attached it to a bathtub for measuring the level of water while filling it. (c) Participants simulated the implementation of a recipe for such scenario: the "water-level" trigger was replaced with the available "temperature" trigger.

Figure

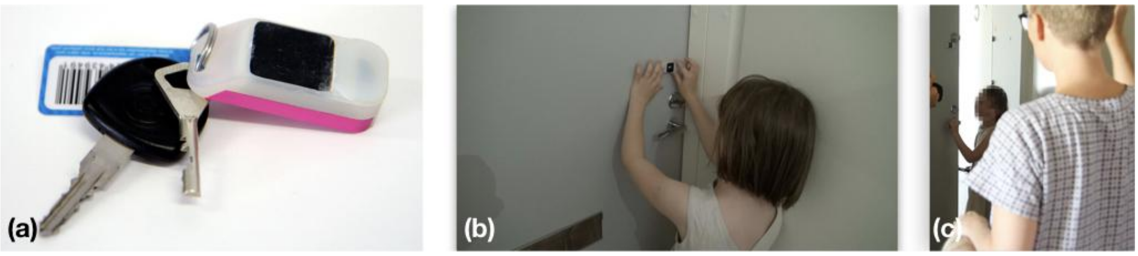


Figure 11. Partial implementation of a scenario for being notified to pick up the keys and/or the smartphone when leaving the house (Safety high-level goal). (a) Attaching the token to the key ring. (b) Attaching the token to the main door. (c) Opening of the door and triggering a notification.

Figure

Figure 12: Participants acting out a scenario for automatically switching on and off the veranda light. (a) Attaching the token to the veranda door. (b) Opening and (c) closing the veranda door.

Figure

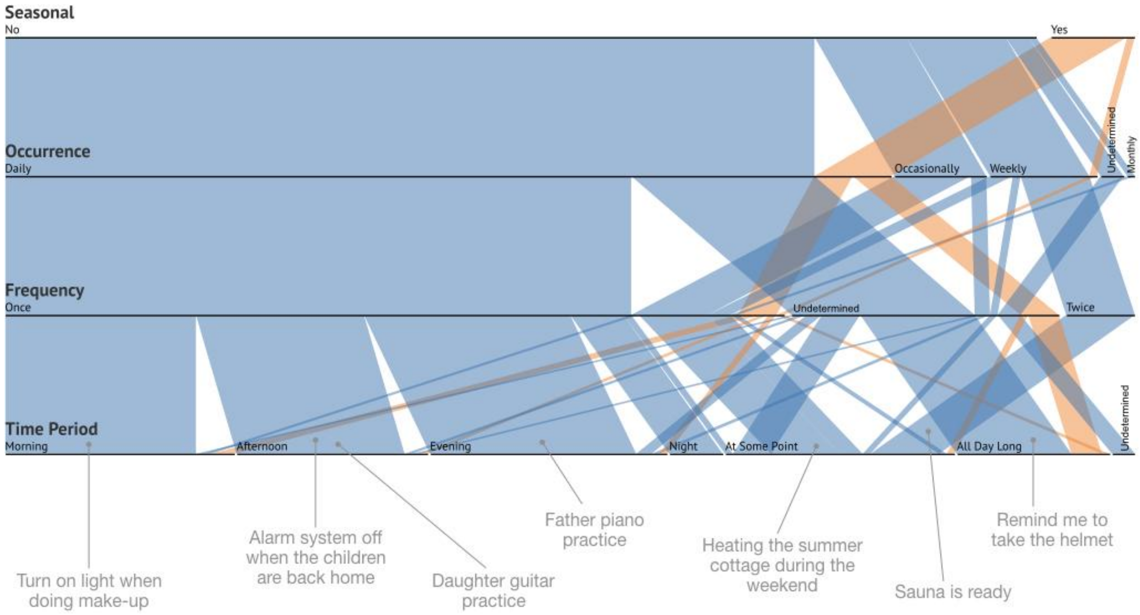


Figure 13: Ephemerality of activities. The parallel set chart shows scenarios' frequency and occurrence. Blue lines show activities that are not seasonal, while red line are used for seasonal activities.

Figure

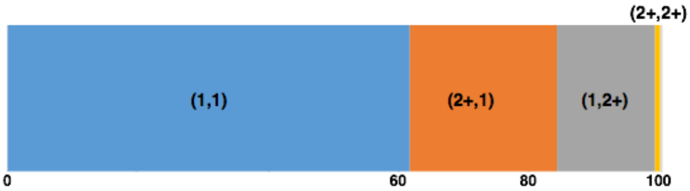


Figure 14: Percentages of trigger and actions combinations. Category names correspond to the number of triggers and actions. For example, (1,1) is basic trigger-action programming: 1 trigger and 1 action. 2+ means two or more triggers.

Figure

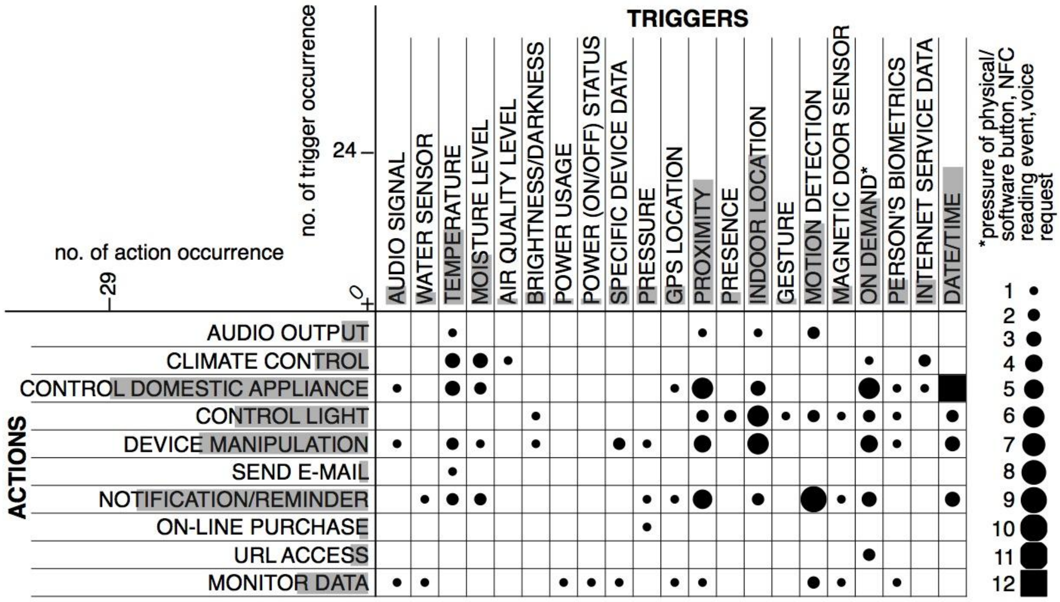


Figure 15: Occurrences of each trigger, action, and their combinations.

Figure

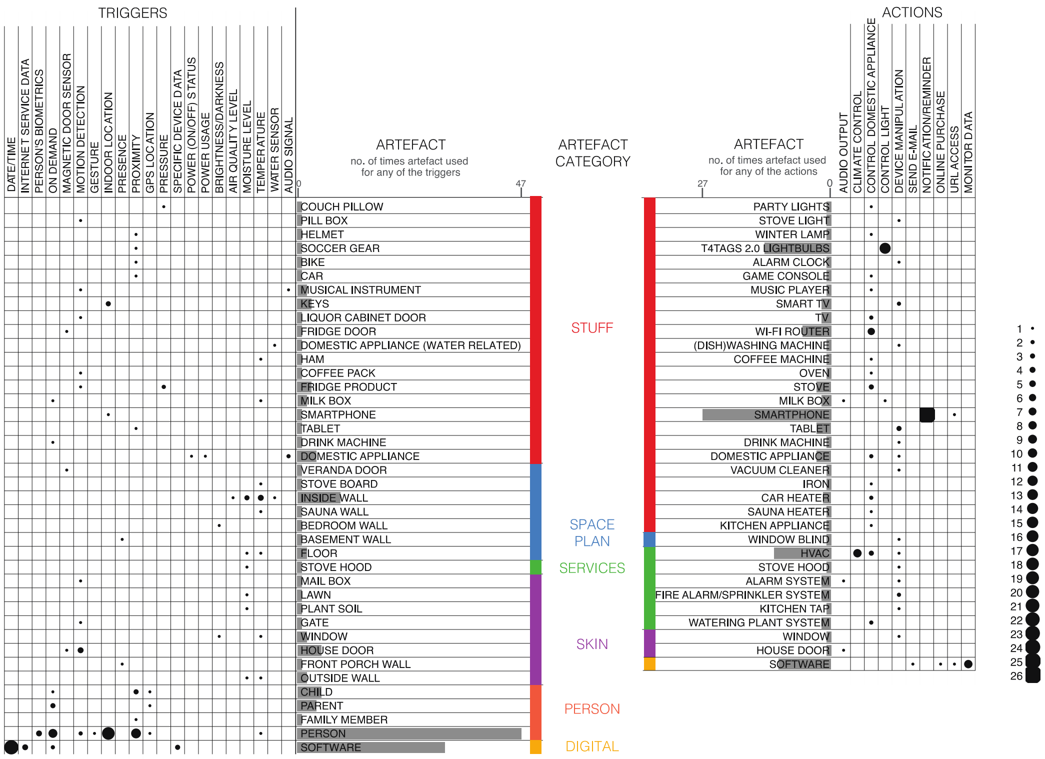


Figure 16: Occurrences of combinations between scenarios' triggers (left side), actions (right side) and the artifacts that can be present in a home environment. The artifacts are organized according to the Shearing Layers concept [Brand 1994].

Figure

Figure 17: (a) Using of a token to sense the child's temperature. (b) Simulating fever by attaching the token to the kettle. (c) Sending an email to the child's school for notifying her absence.

Figure

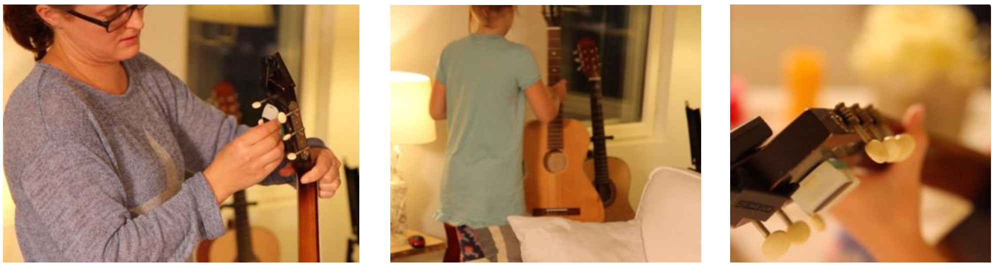


Figure 18: Participants acting out a scenario for checking if the daughter is playing the guitar.