Personalizing a Student Home Behaviour

Luca Corcella, Marco Manca, Fabio Paternò

CNR-ISTI, HIIS Laboratory
Via Moruzzi 1, 56124 Pisa, Italy
{luca.corcella, marco.manca, fabio.paterno}@isti.cnr.it

Abstract. Trigger-Action programming is emerging as an expressive and effective approach when customizing services and applications that have to react to several dynamic events. Recent research efforts aim to overcome some limitations of existing commercial tools in supporting editing of personalization rules. However, they have often been applied and assessed in laboratories. In this work we report on how a personalization platform has been applied to an application controlling the home of a group of students. The home has been equipped with various appliances and sensors accessible through an Arduino platform. The personalization platform has been customized to integrate with the home application through a context manager middleware. The resulting personalization tool and the home application have been used and assessed by the students living in the home and various colleagues and friends without programming experience.

Keywords: End-User Development for home applications, Trigger-Action Programming, Context-aware applications

1 Introduction

Major recent technological trends have seen the advent of mobile and Internet of Things technologies. This has implied that we live in more and more variegated contexts of use characterised by the dynamic association of people, devices, things, and services. Many types of dynamic events are generated in such rich contexts of use. At the same time the number of users accessing applications and interested in their potential customization is increasing [1]. Such users have variegated needs and interests, which often are not well supported by existing applications. Thus, it becomes fundamental to provide users without programming experience with tools that allow them to personalize the context-dependent behaviour of their applications [2]. A typical domain in which such issues often raise is the home. Indeed, various sensors and appliances are being introduced to make our homes more intelligent for security, heating, and other aspects. However, different people may have different views about the most suitable way to exploit such technologies, and it is difficult to find applications able to satisfy all the possible needs.

In recent years such issues have stimulated the interest of various researchers [2] in finding novel solutions to ease the use and customization of context-dependent applications. The goal is to obtain intelligent environments in which both devices and users are able to perceive and interact in the environment. In this perspective End-User Development (EUD) [3] tools can play an important role in transforming users from passive consumers of existing applications into active producers in determining the behaviour of intelligent environments. Various apps for customizing the behaviour of existing applications in mobile devices or Web services have been introduced, such as Atooma[1], Tasker[2], IFTTT[3]. In general, they adopt the trigger-action programming paradigm in which some events and/or conditions determine the performance of some actions. However, such approaches present difficulties to find a good trade-off between usability and expressiveness (the ability to specify the many possible types of triggers and actions) [4].

One tool that aims to overcome such limitations is TARE (Trigger-Action Rule Editor) [5], an editor for specifying trigger-action personalization rules that can be applied to Web applications. In the corresponding personalization platform it is integrated with a context manager that is able to connect to a variety of sensors and devices, and provide logical descriptions of the events and conditions that are verified. However, its evaluation has been carried out in a laboratory, and we deemed it interesting to investigate its use in realistic environments. Thus, we have equipped a students' home with a number of sensors and devices with the support of an Arduino board. The home is a typical environment suitable for the deployment of sensors and actuator representative of the Internet of Things possibilities. We chose a students' home because they showed interest and enthusiasm in participating in the experiment, in addition, they share common rooms in the house, and this can highlight particular situations that can hardly occur in other homes. We have customized the context manager for accessing the sensors and devices in the considered home, and a responsive Web application has been developed to provide users with easy access to the appliances' state and control it. Then, the Web application has been integrated with TARE in order to allow users to define personalization rules and check the effects of their performance. We have then carried out a usability test to assess the effectiveness and usefulness of the resulting personalization environment and its application, and gathered feedback from the students after use over long time. To summarize, the contributions of this work are:

- Demonstrate how a general personalization platform can be integrated in a real context of use (with specific sensors, devices, and appliances) and an associated application;
- Report on the usability of the integrated personalization platform and application assessed through a user test carried out in the actual context of use, and trials carried out, still in the real context, over a one month period.

In the paper after discussing related work, we present some background information on TARE, next we describe the home application developed and the students home equipped with various sensors and devices. Then, we illustrate how they have

[1] https://play.google.com/store/apps/details?id=com.atooma
[2] http://tasker.dinglisch.net
[3] https://ifttt.com/

been integrated with TARE and the associated personalization platform, and report on the usability test and discuss its results along with the experiences of the home inhabitants. Lastly, we draw some conclusions and provide indications for future work.

2 Related Work

In recent years various apps have been introduced to support some level of personalization. They have different features and complexity. Tasker requires some technical knowledge, while Spacebrew[4] users should have some knowledge in JavaScript programming. Atooma and IFTTT are free tools and more intuitive in terms of use, but still with some limitations. However, such tools do not provide immediate support for home personalization. In this perspective an interesting contribution is AppGate [6]. It is composed of a set of tools that support the specification of rules, which are described through a subset of natural language by an editor that aims to support their specification taking into account the actual state of the home context. A first test of this tool has been carried out in the home of the researchers who designed it and some volunteers. The users specified some rules, indicating 3-5 that on average can be useful in their daily life.

IFTTT is an environment that allows users to connect existing Web applications (such as Gmail, Twitter, Facebook) and devices (such as Philips HUE, Nike+, ThermoSmart, Samsung HUB). Each rule can contain only one trigger and one action. Its limitations are that it does not support rules with trigger compositions and the list of applications that can be connected is not easy to manage and understand. A recent study [7] indicated that users can quickly learn rules with multiple triggers and actions. Atooma is an Android app that allows users to specify rules, with also the possibility to include up to five elements in the triggers and actions. The possible triggers and actions are grouped in categories in order to facilitate their identification and access, even if they are less than those supported by IFTTT. A study [8] reported a user test with these two applications with some qualitative and quantitative results. They seem to indicate that users liked more Atooma than IFTTT, the rule-based approach has been appreciated, also because it allows them to easily automatize and personalize some daily activities.

Overall, the studies that have addressed usability of trigger-action programming tools have often been carried out in laboratories (for example [8]), far from realistic contexts of use where to immediately perceive the results of their execution. One exception is AppGate, which was tested by the authors in their home. In this work we present a study that assess in a real context of use the usability of a rule-based personalization platform, which can be applied not only to the home but also in other environments such as shops, hospitals, and data warehouses.

[4] http://docs.spacebrew.cc/

3 Home application

In order to test the personalization environment a responsive Web home application has been developed, which allows users to customize and control the home appliances and sensors and check their status (Figure 1). The sensors are able to detect the temperature, humidity, gas and smoke presence, and the motion; moreover, in the house there are some appliances that can be remotely controlled (tv, radio, fan and the entrance, kitchen and living room lights).

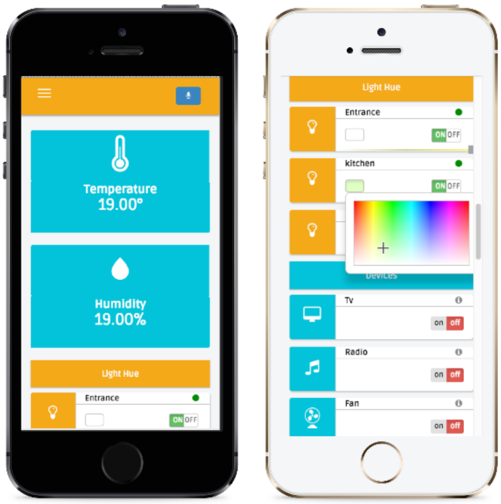

Figure 1: Home application (mobile version)

In order to read the sensors values we used an Arduino board equipped with an Ethernet interface to connect it to the router; moreover, we connected the Arduino board to a series of sensors (such as Kookye Smart Home Sensor Kit for Arduino and Raspberry Pi[5]). Thus, in this case, Arduino acted as an hub, and for this purpose we implemented a RESTful web service, which provides the application with a JSON object containing the values read from the sensors. The backend of the home application has been developed with Node.js[6], an open-source, cross-platform JavaScript runtime environment with an event-driven architecture capable of asynchronous input/output processing. Moreover, the application backend is able also to communicate with the smart objects that are installed in the home (e.g. the Philips HUE lamps) in order to change their state. The user interface is presented as a dashboard and users can immediately interact with the home appliances and visualize the data read from the sensors. Sensors and appliances are organized in a grid layout and represented through the Card component of Material Design. The grid layout along with the Card component facilitate the visualization of the user interface in small devices such as smartphones, where the cards are displayed one below the other.

---

[5] http://kookye.com/2016/08/01/smart-home-sensor-kit-for-arduinoraspberry-pi/
[6] https://nodejs.org

The user interface updates the sensor values through asynchronous AJAX request to the backend part. The application also integrates a vocal interface to interact with appliances and sensors implemented with the JavaScript library Artyom.js. In addition to the management and control of smart devices, the application handles some external services, such as Weather and Calendar Events.

4 The Personalization Approach

TARE has been developed in such a way that users can customize and adapt applications belonging to different domains, such as smart retail, ambient-assisted living, smart homes. Its main goal is to allow users without programming skills to be able to combine, configure, monitor and customize various aspects of an application depending on the context of use. Some characterising features are the possibility of combining multiple triggers and actions; natural language feedback; support of a meta-design approach; and trigger-action rules management.

The editor utilizes a context-model that is supported by the Context-Manager, a middleware component which integrates all the contextual information from the various sensors. The Context Manager is composed of a server and various delegates installed on the available devices. The context model is structured along four dimensions (user, environment, technology and social relationship) which aim to describe the relevant aspects that can affect interactive context-dependent applications. The User dimension describes all the information related to the user such as personal data, education, preferences, position, etc. The Environment dimension defines all the characteristics of the space where the application is executed or where the user acts. The Technology dimension considers all the attributes of the devices, sensors and appliances that are present in the considered context of use. Finally, the Social dimension concerns the social relationships that can exist between people that are present in the context of use.

Our starting point was a general context model that had to be customized in order to create a more specific one to consider all the relevant aspects in the target home. For this purpose, we identified the sensors installed in the home and the available objects and devices in the home. A new Context Model Editor has been developed in order to facilitate the refinement of the generic context model. It is an interactive software able to graphically show the generic context model in a tree-like representation and support its editing. The root is the context node which is decomposed into 4 categories representing the four main dimensions. Each node in the tree represents a context category which is further decomposed into other categories or entities. Three different colours have been used: blue for dimension nodes, green for the categories and pink for the context entities (Figure 2). The context model editor does not require any programming knowledge. It should be used by some domain expert that is able to clearly indicate the relevant contextual aspects.

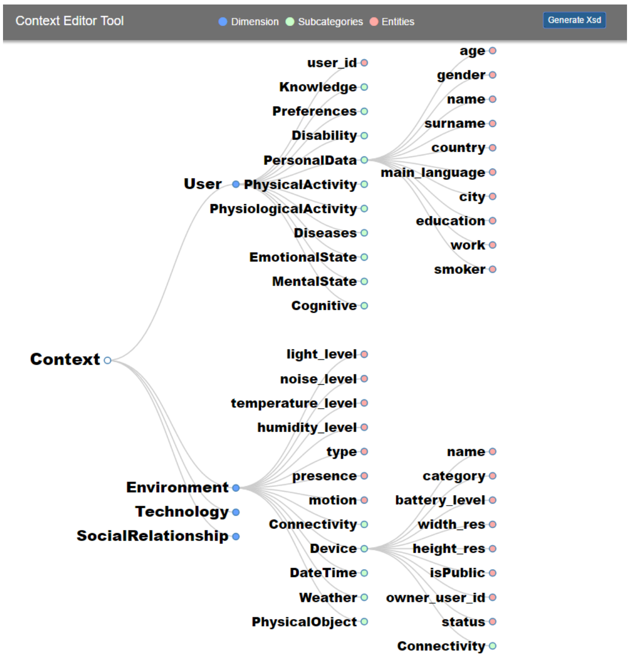

Figure 2: Context-Model Editor

When all the changes on the generic context model are done, it is possible to generate automatically an XML schema with the description of the specific context model. Such schema can be loaded and interpreted by the Trigger-Action Rule Editor (TARE), which will present all the corresponding triggers that can be useful to define the personalization rules. In order to create a TARE instance specific for the target context of use it is also necessary to indicate the smart objects, devices and appliances that exist in it.

The Trigger Action Rule Editor (TARE)

TARE is a Web tool that helps users without programming experience to create personalization rules in order to customize the behaviour of devices and applications. Users can start creating rules either by triggers or by actions and it is possible to reuse existing rules as a starting point to create new ones. Rules are composed of two parts: trigger and actions and the basic structure is as follows:

IF/WHEN <trigger_expression> DO <action_expression>

The *trigger_expression* part describes the events or the conditions that trigger the rule execution, while the *action_expression* defines the actions that should be executed. Each expression can be composed of one or more triggers/actions: triggers can be combined through the AND/OR operators, while actions are executed sequentially.

During the rule editing, triggers are defined by navigating through the hierarchy described in the context model defined through the Context Model editor: each contextual dimension is traversed passing through a number of conceptual levels until a context entity is reached (the context entities are the basic elements of the context model). In order to show only the relevant elements, the contextual levels are presented in an interactive way: the editor only shows the children of the selected element. Figure 3 shows the TARE interface representing the editing of a context entity, the yellow rectangles represent the path to reach the selected entity: Environment -> Physical Object -> hue lamp 1 entrance -> State.

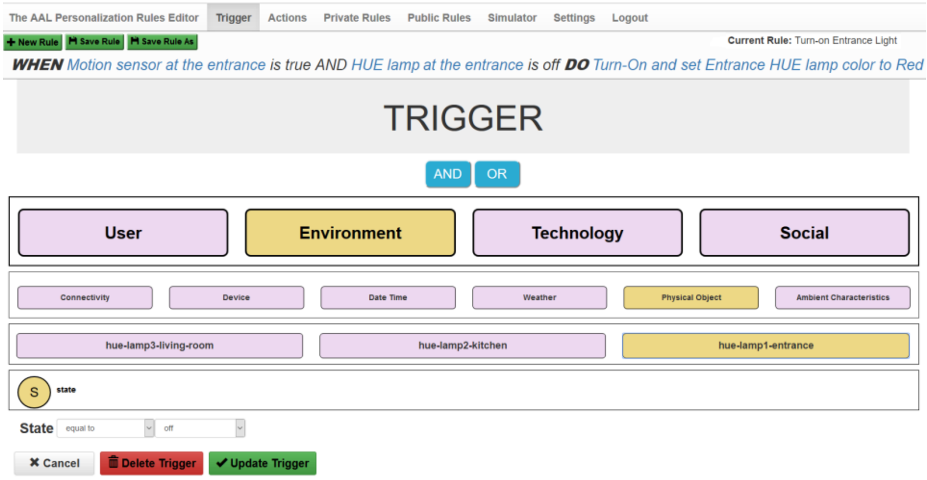

Figure 3: TARE user interface

In the upper part of the tool there is a continuous feedback in natural language indicating the current edited rule. The distinction between events and conditions is highlighted through the keyword IF (condition) and WHEN (event); users can express this distinction by selecting the operator IS (condition) or BECOMES (event).

When the Authoring tool is configured users can start to edit the rules for the target context of use. Rules are saved in JSON format and then sent to the Adaptation Engine module. When the application is deployed it subscribes to this module (Adaptation Engine), which is in charge of subscribing to the context manager in order that it be informed of the occurrence of the events and conditions defined in the rules. Then, when a trigger is verified the application receives, interprets and applies the actions of the rules indicating how to modify the state of the involved devices and appliances.

5 Integration of the Home App and Personalization Platform

As a first step, the house environment was studied in order to understand which sensors can be useful, where to put them and how to access them. A students' home was chosen to bring the personalization environment out of the laboratory and evaluate it in a real case. The selected house is composed of 4 bedrooms, a living room, a kitchen and a bathroom; it is occupied by 4 people including 3 university students (not computer experts and without programming experience).

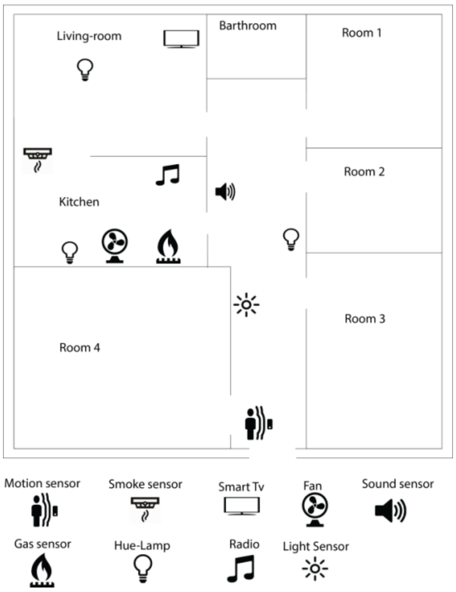

Figure 4: Map of the student house with sensors and devices

The people living in the house chose to install sensors only in common areas, which are: kitchen, entrance and living room. The goal of the home application was to increase the home comfort; the level of security within the home; entertainment; and monitor the domestic consumption. An Arduino board was used to connect with the sensors, and it was expanded with an Ethernet interface in order to connect it to the router and then to the internet in order to provide data to the application which was located in an external server. Figure 4 shows where the sensors were installed within the house: at the entrance it was positioned the motion, noise and light sensors and a Philips hue lamp; the Arduino and Philips bridges were connected to the router installed at the entrance. The Philips Bridge was able to communicate with the three Hue Lamps installed in the house through the ZigBee protocol.

The kitchen was equipped with the smoke and gas sensors, a Hue lamp, and two electric smart sockets to which the fan and the radio devices were connected. These smart sockets were connected with Arduino, which is able to provide the electric power when needed. The television and a Hue lamp were installed in the living room.

In the domotic domain, users personalize and control their devices though applications which are able to manage some objects and visualize the data originated by the sensors. Usually such applications provide a limited set of customizations, and they are not able to react to various events that can occur in the context of use. The personalization platform can be integrated with the applications in order to provide more extended support for this purpose. Thus, the home application described has been integrated with the Personalization Platform in order to allow users to further personalize the application so that it can dynamically reacts to a broader set of contextual changes according to the users' indications expressed with the trigger-action rules.

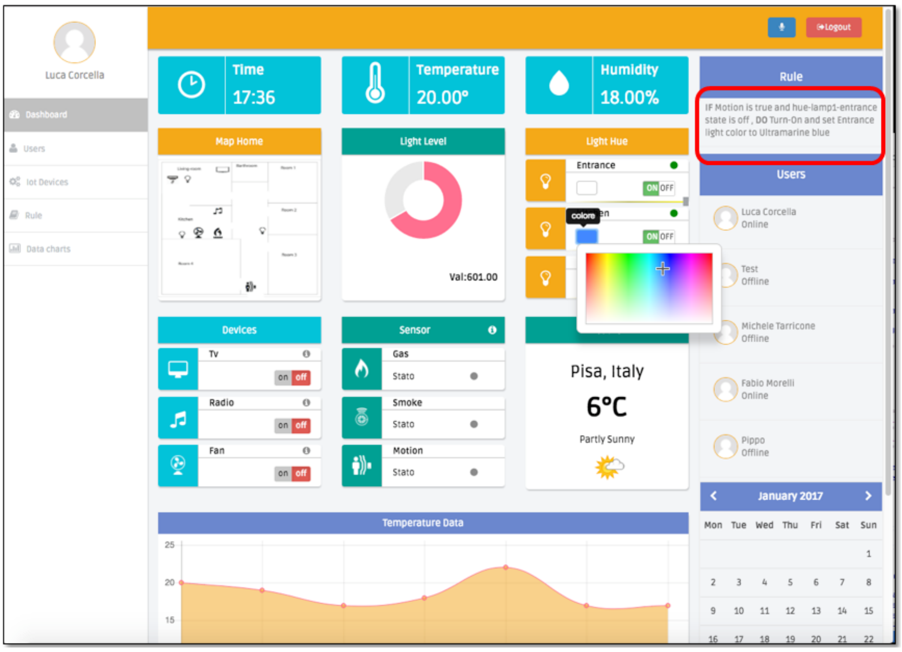

Figure 5: Domotic application integrated with the Personalization Platform

For implementing the integration, the application has been extended with a script able to interpret and visualize the personalization rules created with TARE. Thus, the home application can receive the list of rules associated with it and subscribe to the Adaptation Engine in order to receive the corresponding actions when a relevant rule is triggered. These rules can be visualized in a panel added on the right sidebar (Figure 5 shows an example where one rule is visualized in the top-right part).

Figure 6 shows how the server side application modules communicate with the Personalization Platform. First of all, when the home application is deployed the server side module (*Subscription Module*) subscribes to the Adaptation Engine indicating its name and the application end point through a REST service. The application end point is the URL of an application service that will receive the actions to perform from the Adaptation Engine. Then, if a user specified a rule such as: “When the light level in the kitchen is less than 50 then turn-on the kitchen light” and the light level actually goes below this threshold, then the Adaptation Engine will send the action to the application end point specified during the subscription. In order to trigger a rule, the Context manager should be informed every time a sensor, a device or an appliance change their values: e.g. when a flat mate turns on a light (through the application or through a physical switch) or when motion sensor detects a movement then the IoT manager sends these updates to the context manager. The Context manager exposes a RESTful web service for each context entity that should be updated. There is a module (*IoT Manager*) on the application server side part in charge of monitoring the IoT sensors installed in the house. Each time a sensor or an appliance changes its state, this module will update the Context server through the corresponding REST service.

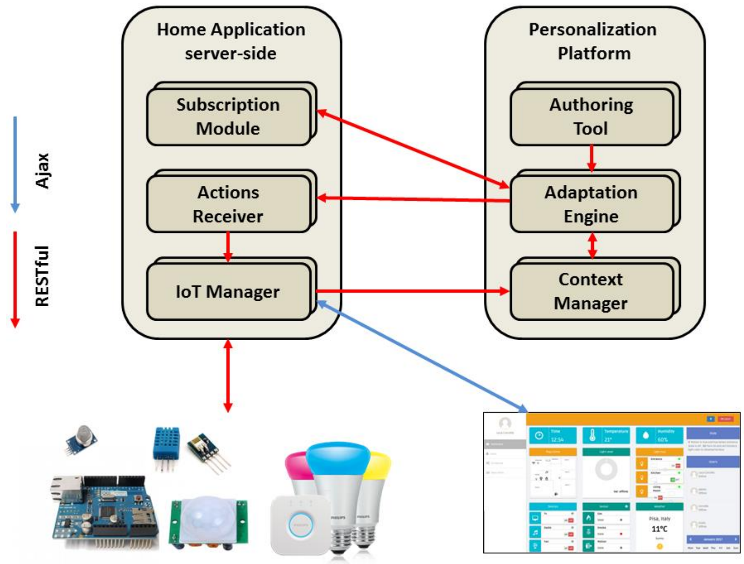

Figure 6: Integration between Domotic Application and Personalization Platform

Finally, there is the module (*Actions Receiver)*, which corresponds to the end-point specified during the subscription. This module receives the actions when a rule is triggered, interprets them and communicates with the IoT manager in order to change the state of the appliances involved in the actions. There can be two different types of actions: Update, which will update the state of a home device, thus the Action Receiver module will call the corresponding function, implemented in the IoT Manager module, specifying the involved device, the action type (change state, change colour, etc.) and the new value (e.g. state = on, colour = red); Invoke Function, this action specifies which function the application should invoke, examples are: sending a notification or a SMS, or displaying the charts of the temperature, humidity or power consumption.

6 Usability Evaluation

We report on the use of the application and the personalization tool in a specific test and over a one month trial period. For the test, a number of additional users were invited in the students' home in order to receive feedback from a broader audience still in the same real context where the actual effects of their personalization could be observed. We think that this type of feedback is more meaningful than that obtainable in a laboratory because in this case users can have a better and immediate understanding of the actual effects of the rules that they specify.

Usability Test

The test was carried out by twenty users (12 males), aged between 20 and 28. They had some familiarity with technologies, in particular Web technologies, but not experience in programming. In order to access the various features of the application, the tasks they had to perform were:

1) Turn on the lights of the living room and the kitchen.

2) Turn on all the lights and set the indicated colours: kitchen (yellow), living room (red), entrance (blue).

3) Navigate in the application to find the weekly data on electricity consumption, humidity and temperature.

4) Turn off the lights by voice command ("turn off all the lights"). Once the command was uttered, click the microphone icon to actually run it.

5) Turn on the TV and the radio.

6) Create this rule: IF hue-LAMP3 is on and TV is off DO turn on TV.

Then, they had to create personalization rules with the following structures and involving the contextual aspects and action types indicated, such tasks were chosen in such a way to be of increasing complexity and address the various relevant contextual aspects:

7) Simple Trigger (technology) + simple action (alert).

8) Trigger compound (environment + user) action + compound (alert).

9) Trigger compound (environment + environment) + simple action (functionalities).

10) Trigger compound (environment + technology) + actions compound (appliances, functionality).

11) Trigger compound (user) + action compound (appliances, reminders).

12) Trigger compound (user environment + technology) + simple action (appliance).

13) Create a new rule from a rule created earlier.

Participants received an introduction to the motivations and goals of the test, and a high level description of the application, TARE, and the structure and possible content of the rules. The users were observed by one of the authors during the test in order to annotate particular comments and emotional states, their sessions were also logged with Camtasia.

Figure 7 shows a box-and-whisker plot reporting the average task performance time along with minimum, maximum and quartiles. Users seemed able to interact comfortably with the sensors and appliances, easily understanding how they were represented in the application user interface. The task performance with the personalization tool (tasks 6 to 13) was variable. Most users seemed to perform better over time. Simple rules required less time. Task 8 took relatively long, probably because it was the first example where they had to compose two triggers. Some users encountered some difficulties in understanding the use of the logical operators to compose triggers and tried to avoid their use. Some users thought that they could use the logical operators even in the action specifications. The data indicate that users became more familiar with editing the rules over time and became more efficient. Indeed, if we consider the rule associated with task 12, it involved three triggers and one action with an average time of 2’24”, while trigger 8, which required only one trigger and one action took on average 2’36”.

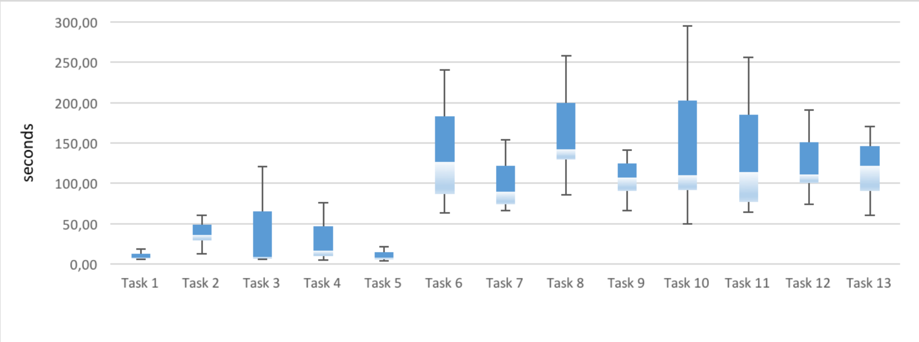

Figure 7: Task completion time

Regarding the application, task 3 and 4 presented some problems for some users. Task 3 was intended to verify whether the search for information regarding weekly electricity consumption, the house temperature and humidity. Its box plot indicates that a few users encountered some problems in finding the requested data: they did not notice the link to access them, and thus navigated across the application for a while before finding them. Task 4 related to the use of voice commands to interact with the sensors. The command to activate voice input was not well positioned in the user interface, and therefore many users sometimes took some time to find it and often asked for help from the evaluator.

We found three types of errors: wrong selections of the elements defining the triggers or the actions; wrong ways to compose triggers and actions or use the logical operators; wrong selections in the application, for example in selecting the page where the electrical consumption or the temperature are reported. Some errors were performed also in the use of the vocal commands because some users did not understand that it had to be explicitly activated.

	Simple rule	Complex rules
Executable	17	71
Not executable	1	5
Not correctly executable	2	24

Table 1: Types of error occurred during the test

In order to perform tasks 7 to 12 users specified 120 rules. The rules after the specification were executed so that users could see their effects in the home.

We can notice (Table 1) that 88 were correctly specified and executed. In the analysis of the errors we have classified them into two categories: moderate and serious errors. Moderate errors refer to wrong values in a trigger definition or a wrong parameter in action definition. The resulting rules can still work but the specified behaviour is not exactly the desired one. They happened in 26 cases (row three in Table 1). For example, a user specified a rule “IF temperature = 14 DO send a message “too cold” to 3281234567”. While the correct condition should have been “IF temperature <= 14”. Serious errors lead to rules that do not produce any meaningful effect (they happened in 6 cases). An example is “IF there is someone at home and lights are on then turn on the lights”.

The users also had to rate some aspects of the application and the personalization tool through a 1-5 Likert Scale, where 5 was the highest ranking. For the application they rated:

- User interface consistency in the sensors’ description (min: 3, max: 5, med:4).
- Usefulness of voice commands (min: 2, max: 5, med:4).
- Usefulness of the map representing the sensors in the house (min: 2, max: 5, med:4).
- Usefulness of sensor monitoring and intelligent devices control through the application (min: 3, max: 5, med:4)
- Quality of sensor integration with the Home application, and their control (min: 4, max: 5, med:4).

The diagram below (Figure 8) shows the distribution of the ratings. It is possible to note that none of the aspects was rated 1, and there were few 2s and 3s. The vocal interaction was the aspect that raised more concerns, probably because users were not used to it.

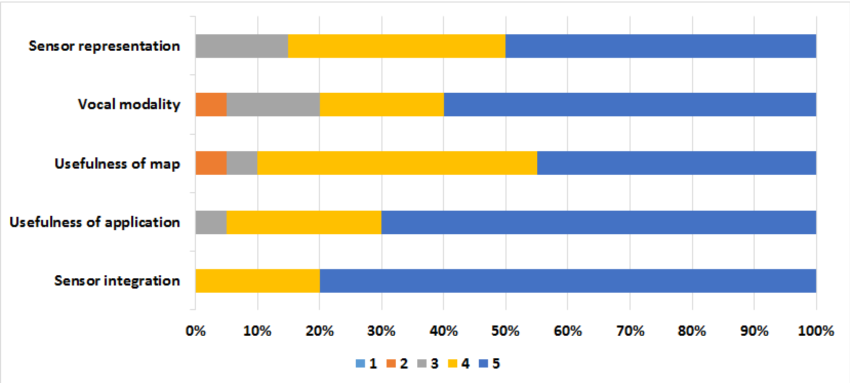

Figure 8: Application Ratings

For the personalization tool the aspects they had to rate were:

- Utility to create rules to customize your smart home (min: 1, max: 5, med:4).
- Quality of the natural language feedback describing the rules create (min: 1, max: 5, med:4).
- Trigger Selection (min: 2, max: 5, med:4).

- Action Selecting (min: 2, max: 5, med:4).
- Distinction between categories and entities of the application (min: 1, max: 5, med:2).
- Rule execution (min: 2, max: 5, med:4).
- Appearance of the Tool (min: 1, max: 5, med:4).

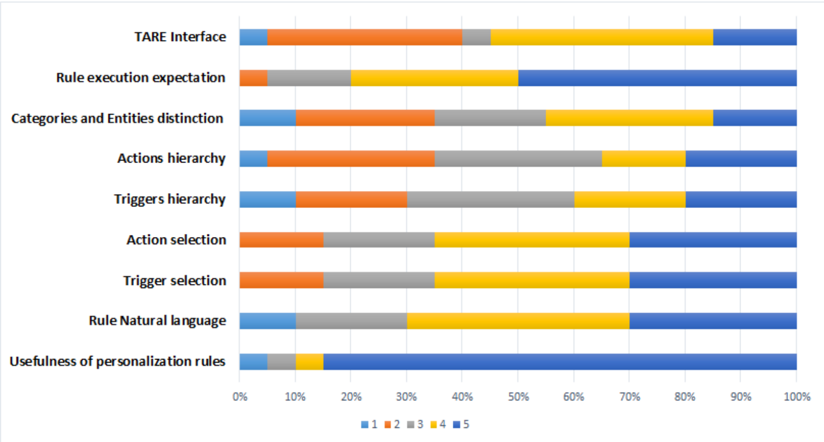

Figure 9: TARE Ratings

Figure 9 describes the distribution of the ratings given to the TARE tool. The results are still positive, even if users reported some difficulties in the part of the trigger selection and actions and the graphical interface of the tool has received less positive scores. As for the problems in finding the trigger attributes, we observed that the majority of users started to look for the desired attribute following a path that made sense for them, that is, choosing a dimension (for example, user), and then continue exploring the trigger structure. The users who were not able to quickly find the desired attribute can be divided into two categories: the curious, that iteratively explored all the possible triggers with the goal of finding at a certain point the desired one; and those who repeatedly tried under the wrong dimension the desired intermediate nodes. In order to complete the task, the evaluator had to ensure that users considered all trigger types required by the rule structure specified in the proposed task. One useful option for those users who tend to "give up" after a couple of fruitless searching could be to provide some automatic support. For example, the ability to specify a keyword (for example the name of an attribute) on a search engine to obtain the path to the corresponding attribute in the contextual triggers structure.

In conclusion, the participants also expressed suggestions and observations by answering some open questions. Regarding further application functionality, they suggested to integrate further home appliances such as heaters, washing machines, dishwashers, etc. In terms of suggestions for the application they indicated a preference for including support for other languages, and to better integrate voice commands, they would have liked that the voice assistant be always active. Regarding the personalization tool, a user suggested the possibility to introduce the ability to create rules by using a form, since he found that the current organization of the triggers and actions is uncomfortable because of the continuous page scrolling that users have to carry out. In addition, some participants suggested to integrate a search engine for triggers and actions. Others suggested the possibility of including a map of entities and categories so that users can quickly orient themselves when searching them.

Trial Report

In addition to this user test we have also collected the results of the experience of the students living in the home who used the application and the personalization tool for a long period of time. One of the authors was able to briefly talk to them and observe their use of the personalized application almost daily for one month. Three of them had no programming experience. In the beginning they were not convinced of the utility of the approach while over time they appreciated it and used it with some satisfaction. They preferred to use the mobile version of the responsive application. After some days of trials with the application they were requested to write some personalization rules that they deemed useful. Examples of the rules provided are:

- WHEN light-level less than 10 and time is 22:00 and entrance light is off DO Turn on Entrance Light.
- WHEN time is 22:00 and tv is off and motion is true, DO turn on living room tv.
- WHEN gas sensor is true, DO send alarm to 3281234567.
- IF tv is on and radio is on and fan is on DO show consumption data.
- WHEN time is 09:00 and appointment type is wake-up DO Turn on Kitchen Radio.
- WHEN Temperature is more than 20 and Fan is off DO turn on Fan.

An important observation derived from the trial in a real context that also opens a point of interest for future development efforts concerns the management of conflicting rules specified by several users for the same context. For example, the following rules were in conflict: i) In the morning lights should be off; ii) When user is at home lights should be on. These rules may conflict depending on when the user returns home and the result of the execution of the first rule can be cancelled by the execution of the second one.

In the home another issue emerged related to the rules created by different users, which were problematic to execute since users of the same environment may have different habits and needs, and the rules created by one person may not satisfy another person's needs or habits. Such rules are not in conflict because they manipulate the same resources (as the previous example with the lights) but they should not be performed since they do not meet the needs of all inhabitants. For example, it happened that a flatmate specified a rule whose execution created problems because it conflicted with the needs of the other housemates. This situation occurred with the following rule: - WHEN time is 8:30 AM DO Turn-on radio.

In this case the radio would wake up the housemates who want to continue to sleep at that time, while the user specified this rule in order to listen to the radio, which is located in the kitchen, during breakfast.

7 Conclusions and future work

In this paper a solution that allows end-users to customize applications taking into account the context of use by specifying trigger-action rules has been presented. In order to assess the effectiveness of this approach, the TARE tool and the corresponding platform have been customized to a specific home context of use. A home application has been developed and has been integrated with the personalization platform in order to test this environment in a real context.

The performed user tests indicate that people without programming experience are able to define personalization rules through the TARE tool with limited effort, and can learn how to use it in short time. The reported experience consisting in customizing and deploying the personalization platform for a students' home highlighted the usefulness of the proposed approach in a real context. It also indicated some issues that were not considered sufficiently before: sometimes the execution of the rules can produce effects that may not match the needs of all the users in the considered context. Future work will be dedicated to improving the TARE tool in order to better reasoning about rules performance in multi-users environments.

References

1. Bellucci, A., Vianello, A., Florack, Y., & Jacucci, G. 2016. Supporting the Serendipitous Use of Domestic Technologies. IEEE Pervasive Computing, 15(2), 16-25.
2. Desolda G., Ardito C., and Matera M. (2017). End-User Development for the Internet of Things: EFESTO and the 5W Composition Paradigm. In Rapid Mashup Development Tools: Second International Rapid Mashup Challenge, RMC 2016, Lugano, Switzerland, June 6, 2016, Revised Selected Papers, Daniel F. and Gaedke M. (Eds.) Springer International Publishing, Cham, 74-93.
3. Lieberman, H., Paternò, F., Klann, M., Wulf, V.: End-user development: An emerging paradigm. In: End-user Development. Human- Computer Interaction Series, pp. 1–8. Springer (2006). DOI: 10.1007/1-4020-5386-X_1
4. Lucci, G., Paternò, F.: Understanding End-User Development of Context-Dependent Applications in Smartphones. HCSE 2014: 182-198, LNCS Springer Verlag. DOI: 10.1007/978-3-662-44811-3_11
5. Ghiani, G., Manca, M., Paternò, F., Santoro, C.: Personalization of Context-dependent Applications through Trigger-Action Rules. ACM Transactions on Computer-Human Interaction, Vol.24, issue 2, 2017.
6. Coutaz, J., Crowley, J. L.: A First-Person Experience with End-User Development for Smart Homes. In: *IEEE Pervasive Computing* 15, 2 (April 2016), 26-39. DOI: 10.1109/MPRV.2016.24
7. Ur, B., McManus, E., Ho, M. P. Y., Littman, M. L.: Practical trigger-action programming in the smart home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 803-812. DOI: 10.1145/2556288.2557420
8. Cabitza, F., Fogli, D., Lanzilotti, R., Piccinno, A.: Rule-based tools for the configuration of ambient intelligence systems: a comparative user study. In Springer Science+Business Media New York 2016. DOI: doi:10.1007/s11042-016-3511-2 .