Deployment and Analytics for Personalization Rules in Internet of Things Scenarios

Marco Manca, Fabio Paternò, Carmen Santoro
CNR-ISTI, HIIS Laboratory, Pisa, Italy

ABSTRACT

This paper presents how the TAREME (Trigger-Action Rule Editing, Monitoring, Executing) platform provides support for executing and analysing personalized automations in Internet of Things scenarios. The platform allows creation and execution of trigger-action personalization rules that can change the state of connected smart objects and devices, send alarms or reminders, and modify applications' state depending on contextual events. The platform is able to support analytics about the actual use of the rules and provide associated information, which can be useful to better understand users' personalization needs. Such features have been deployed in six trials, which have shown the feasibility of the approach and reported fruitful feedback.

KEYWORDS

Personalization Analytics; Trigger-Action Programming, Internet of Things solutions to support end-user development in daily life

1 Introduction

With the spread of the Internet of Things (IoT) paradigm, users increasingly interact with many Internet-enabled objects in a wide variety of diverse contexts, through different applications, for various goals. Moreover, each smart object is typically provided with its controller and application, thus making the IoT ecosystem scattered, and the exchange of data between objects manufactured by different producers problematic. In this situation, the overall behaviour of context-dependent applications cannot be hardcoded by developers at design time, since they cannot foresee all the possible scenarios in which applications would be exploited or whether the results will actually be meaningful to end users.

It has been recognised that software engineering is no longer solely applicable to professional software engineers [7]. Anyone who develops software, whether as a professional developer or not, may well need to employ suitable processes to help ensure that the software they develop is of sufficiently high quality for the purpose for which they intended it. In this regard, several studies have investigated the effectiveness of End-User Development (EUD) techniques for creating *personalized* applications in various domains such as rehabilitation [38], robotics [24, 44], finance [16], Internet of Things [5], smartphones [34], smart homes [9], visualisations of smart home data [10], cultural heritage [3]. However, designing approaches making end users actively involved in the development of solutions to their problems is not a straightforward process, because often there are several design trade-offs [2] that need to be dealt with to identify a high-quality solution in a specific context.

In the area of IoT applications, the approaches exploiting the *trigger-action paradigm* [22, 41] seem to be particularly suitable because they capture well the behaviour of smart objects and can be more easily understood by people without programming knowledge. Indeed, to use this paradigm, there is no need for specific algorithmic abilities to create rules indicating the desired effects (in terms of changes of state of appliances, devices, and user interfaces) when specific situations occur in the actual context of use. The trigger-action approach has also raised interest in commercial tools. One example is IFTTT (If This Then

That)[1], which already dating back in early 2017 had more than 320 thousand automation scripts ("applets") offered by more than 400 service providers, was installed more than 20 million times, and more than half of such IFTTT services were IoT device-related [29].

However, various studies pointed out that there are cases in which the rules that are created by users do not lead to the desired behaviour [45], thereby some work highlighted the need to better support users in this regard, e.g. by providing debugging features [12, 19, 31]. One issue is the distinction between events and conditions [22], which sometimes is not clearly understood. In other cases, the result of the execution of a rule does not correspond to that expected by the user. For example, people often expect that the execution of the rule "If I am in the kitchen, the light should be on" would also implicitly include switching off the kitchen light when the user is no longer in the kitchen, which is not always the case. Unfortunately, various research efforts addressing such issues just remained at a rather preliminary stage: while they show the feasibility of the underlying concepts, they have not been integrated into truly working systems allowing the actual triggering and execution of specified rules in real scenarios. Therefore, their potential effectiveness has been mainly considered in laboratory studies.

Another issue is that, in order to easily make sense of preferred user activities, interests, and routines, the actual use of such trigger-action rules needs to be analysed. However, a 'manual' analysis does not seem a viable approach because the number of rules created by users can be very large. Thus, there is the need to provide effective, interactive, visual analytics solutions facilitating a more intuitive understanding of the kind of personalization activities users are most interested in, improving the design of both the EUD tools and the associated applications, thereby ultimately facilitating their adoption. However, the analysis of the rules created by users can only shed partial light on the personalization activities users are actually considering for their needs in relevant contexts. Users can create rules for different goals and disparate, even temporary, situations. For instance, they can slightly edit the same rule multiple times to save (and possibly test) different versions until they achieve a specific desired effect; they can create rules just to achieve familiarity with the environment; they can even import rules from external repositories just to analyse them, and to possibly reuse them in the future. Therefore, if not thoroughly managed, users' rule repositories could quickly become difficult to analyse, and not all the contained rules will truly reflect actual users' personalization needs. Thus, to get a more relevant and consistent picture of the tailoring activities users are interested in, we judged it important to support the possibility of focusing on rules that have actually been sent for execution, and provide support for their analysis.

In this paper, we present a contribution describing a platform that supports executing and analysing trigger-action personalization rules. Execution of rules is supported by a middleware infrastructure able to manage and detect when contextual triggers generated by available sensors and devices are verified. Analytics is supported through a tool providing information about the personalization rules executed in users' contexts, which can help in better understanding their personalization preferences. In order to better illustrate the features of the platform and the aspects associated with its deployment in real-world settings, we also report our experience in six field trials in the Ambient-Assisted Living (AAL) domain, where we involved ageing people and their caregivers in using the system. It is worth pointing out that the goal of such initial trials was not to improve the health of the participants, but to investigate the feasibility of deploying the platform in a real-world scenario and to more concretely show its features and capabilities in real-world use. Besides, it is worth noting that the personalization platform is not domain-dependent, and can be applied to different types of scenarios. This platform also includes a Tailoring Environment (whose main functionalities have been introduced in [17]) that enables end users to specify expressive trigger-action rules with various possible compositions of triggers and actions, and with a clear distinction between events and conditions defining triggers, in a way that is understandable by end users.

[1] https://ifttt.com/create

In this paper, after introducing the motivations of the research and discuss different target scenarios in various application domains to show the usefulness of the platform in multiple sectors (Section 2), we analyse the state of the art in the field (Section 3), then we describe relevant background information to better define the focus and the scope of this research (Section 4). Next (in Section 5), we describe the platform, by discussing its components and associated communications, with a particular focus on illustrating how this platform enables the actual execution of rules in real, multiple settings. We then present the analytics functionalities, which allow relevant stakeholders (e.g. in the AAL domain they can be the caregivers) to analyse the personalised behaviour triggered in different end users' contexts. Then (Section 6), we report on an instance of the platform that has been deployed in the AAL domain for supporting six field trials in elderly's homes, also describing the specific applications, sensors and appliances considered for the deployment. The analysis of such six deployments of the platform has offered interesting indications. We also discuss (Section 7) some more general lessons learnt in designing, implementing, and deploying the presented personalization platform. Lastly, we draw some concluding remarks and indicate directions for future work.

2 Target Scenarios

In this section, we discuss personalization needs in different IoT-related domains in which the platform can be applied, ranging from industry, ambient assisted living, smart retail, to smart home, putting a particular focus on the relevancy of both executing and analysing personalisations in such specific contexts. They introduce target scenarios that the TAREME platform aims to support.

Industry 4.0. Several dynamic events occurring on various appliances, equipment and IoT objects of a smart industry can potentially impact its efficient and safe functioning and, if properly handled, they can translate into cost savings, better safety, continuous process optimization, and increased plant efficiency and profitability. Such events are very heterogeneous and can significantly vary across different factories, because, even within the same brand, there might be factories with different levels of digitalization. Thus, it can be useful to monitor various types of aspects such as: i) the speed at which products are delivered as well as checking their quality; ii) natural resources' consumption (to minimise waste production while increasing water and energy efficiency to better address sustainability issues); iii) potential risks for the operators in the plant, which can be related to safety (e.g. the presence of wet walking surfaces where workers can slip on, tracking the status and location of operators to avoid that isolated operators can fall unnoticed in remote sites of the factory), environmental concerns (e.g. permissible time of exposure at specific levels of noise, cold, heat, pollution, which can even depend on local/national regulations; potential proximity between walking operators and drivers of elevators moving specific equipment within the plant, who could have limited vision of the place), and chemical risks inherent to operations of specific factories. Therefore, factory managers need to customise their systems for effectively handling the various processes according to the surrounding, dynamic context of the factory. An example of customization is when the quality of delivered products is detected as currently not congruous with the specifications requested by the customers. In such situations, the responsible person can decide to adapt the current configuration used in the production line to avoid that the customer downgrades it. Another aspect is that, in various industries (e.g. in the manufacturing sector), continuous operation is of critical importance: in this case, services must be engineered for high availability and fault tolerance to ensure continuity of service. Thus, an example of rule addressing these issues could be that, when the efficiency of a machinery goes below a specific threshold (especially these expected to work continuously), an alarm could be sent to its maintainers (or its suppliers, and even the producers) to notify them that the machine is starting to work inefficiently, so as to optimize maintenance interventions, reduce the occurrence of quality deviations, and avoid production downtime and/or unplanned shutdowns. Another example of customization could be represented by connecting data regarding the current availability of raw material with proper notifications to the suppliers, so that material can be supplied only when needed. In such a domain, also having an overview of the type of rules that have been put in place in a particular period can give relevant insights on the specific issues that need to be faced in the management of the factory (e.g.

maintenance, safety). Another relevant aspect to monitor could be to understand how many rules have actually triggered alarms in the factory.

Ambient Assisted Living. With the rapid growth of the elderly population, there is an increasing need for supporting such people in preserving an independent and healthy lifestyle in their homes rather than through more expensive hospitalization solutions. Nowadays, the wide variety of sensors, actuators, appliances, devices, smart objects and services that are available in the market can suitably support monitoring of elderly as well as sustain them in their daily living activities. In the AAL domain, triggers represent situations/events that caregivers could be interested to know regarding the elderly: e.g. health/cognitive/emotional status, physical/social activity, especially when they are away from them (remote monitoring). However, to be optimal, remote monitoring services and health-related interventions should be strongly personalized to specific individuals' requirements, preferences, abilities and motivations, which can radically differ among the elderly, and even dynamically evolve over time for the same person depending on changing user needs and context-dependent conditions. An example of customization in this scenario can be represented by the possibility of sending specific reminders to elderly (e.g. when they are expected to take a medicine), by using their favourite notification channel and depending on the specific equipment available in their house (e.g. mail, SMS message, notification on a smartphone or a tablet or the TV, vocal reminders through a voice assistant device). Besides, depending on the type of diseases affecting the elderly, a caregiver (or even the elderly) can set up a rule that better monitors what the elderly does during the night, or whether the elderly makes their expected physical and/or cognitive exercises during the day. Moreover, still depending on the specific routines, impairments, and even the layout of the house of the elderly, another rule can switch on a light for a specific interval of time so that the bedroom-bathroom path can be better illuminated when motion is detected during the night. In addition, in such contexts, caregivers can benefit from a rule monitoring tool that allows them to understand which kind of rules have been triggered in a specific period, to better understand the automations that end users preferred to put in place and their goals. For instance, high use of reminders can be interpreted as a sign that the elderly is increasingly relying on external support to remind the various activities to carry out, which might need further investigation in some cases. In addition, in this domain, lights can play an important role in the elderly mood, thus a caregiver can define a rule that changes the light's colour according to the current user's emotional status.

Smart Retail. The retail domain represents another sector in which personalization can offer valuable opportunities. In this area, the same application can be deployed for very different contexts of use, from large supermarkets in malls (typically located within newly built buildings, with highly technological infrastructures that allow easy integration of novel sensors and devices), to smaller shops in towns (thereby having more limited technological opportunities). Thus, the same application associated with such two types of shops has to provide different, relevant, tailored experiences to its customers. For instance, owners of big markets would aim to offer a high-quality mobile shopping experience exploiting advanced technological media and customized services (e.g., dynamic personalization of prices and offers based on current customers' behaviour while moving in the shop, suggestions of products that are complementary to those that users currently have in their shopping cart, personalized offers to returning visitors), also to make the shopping experience more efficient (e.g., real-time directions for locating hard-to-find products, mechanisms to avoid queues). On the other hand, the version targeting smaller shops could not provide specific support for assisting customers while they are in the shop (due to the limited availability of sensing technology), but it could include personalized lists of daily and weekly offers tailored according to e.g. customer profiles, their latest purchases, seasonal aspects, etc. In addition, a retailer can decide to adapt the navigation of the online shop to the habits of their customers, by reducing the number and complexity of the navigation menus or by showing to each customer only the navigation elements best suited to their behaviour history or their profile, so that customers can find their way in the online shop in a faster and easier manner. In this domain, a monitoring tool could be useful in understanding the most used personalizations that various shops have put in place for better support their customers.

Smart Home. The smart home represents another relevant scenario. Indeed, home-related activities are the result of interactions involving different aspects of human life, e.g., work, leisure, health, security, safety, which mainly reflect the inhabitants' needs. These can be routinely, but also temporary and changing over time, or they can be personal but also shared by different members of the family. Therefore, the great variety of activities that occur in the domestic environment and the interactions of people with the devices and objects available in the household emphasize the need of support for enabling easy customization and personalization by end users. An example of customized automation is the creation of a rule that notifies parents when children are back home from school, and also activates the external surveillance system in the house. Another rule could check gas leaks in the kitchen, and in such case, an alarm can be directed to alert inhabitants currently at home (as well as other people that can provide help), and at the same time activates the automatic opening of the kitchen's windows. In this case, a rule monitoring tool can be useful to better understand the kinds of automations that the family members are most interested in, i.e. whether they regard controlling the equipment at home, or whether they involve triggers more related to inhabitants' safety (e.g. presence of smoke or gas) or health (heart rate, time spent in the bathroom).

3 Related Work

Recently, due to the importance of customizing the behaviour of IoT applications, several approaches have been proposed in the EUD area to put the personalization of IoT ecosystems in the hands of end users, who are the stakeholders that are most familiar with the actual needs to support [14, 17]. The approaches that have been used until now exploit a variety of metaphors and programming paradigms [4], e.g. component-based [1], process-driven [31], programming by example [25], natural language [20], tangible [40], and rule-based [15]. In particular, rule-based approaches are receiving increasing interest, since end users can easily reason about how to connect contextual events with corresponding behaviour of their smart objects and applications. One of the best-known examples is IFTTT, where users can create automations by composing online services from different providers, by indicating one trigger and a corresponding action. However, even if rule specification is simpler than specifying blocks of code, rule-based approaches can become difficult for non-programmer users when complex rules have to be expressed. The correct formulation of logical expressions implies knowledge of some key concepts (e.g. Boolean operators, priority of operators) that may not always be intuitive for them. In this regard, despite its popularity, IFTTT itself is not free from limitations, since it only provides support for building basic rules (including one trigger and one action), thereby making it unsuitable for more structured rules [41], i.e. those requiring multiple triggers or actions. Moreover, it does not clearly distinguish between events and conditions. Therefore, further effort to enable end users to specify rules combining multiple triggers and actions should be pursued because this would provide them with the possibility to indicate more flexible behaviours [17, 28]. Some solutions in this area, while managing to show the feasibility of the underlying concepts, have shown their effectiveness mainly in controlled in-lab studies (e.g. [5, 15, 18]), which unfortunately are not able to shed light on the plethora of unpredictable issues that can arise during deployment, installation, operation and maintenance of such systems in real and heterogeneous IoT-based trial environments.

Tetteroo and Markopoulos [36] have already shown that field evaluations of EUD systems are relatively uncommon, and in a next study [37] they highlighted key aspects that make EUD deployments different from regular software deployments. Indeed, only few EUD approaches have shown to be sufficiently robust for their use in longitudinal studies in the wild with real users. In one of the few examples [13], the authors report their own first-person experiences of deploying AppsGate (an EUD tool designed by the authors themselves) in their own home for more than a year. Another example in the rehabilitation domain [38] describes the deployment and adoption of TagTrainer, which was deployed in four rehabilitation clinics and used by 24 therapists. However, this deployment was done in rather controlled environments (clinics) for supporting very specific tasks, whereas the deployments we report here have been conducted in private homes, having different configurations and users' requirements.

In addition, the scarcity of truly working personalization solutions exploited in real settings could have also affected the scarcity of tools allowing relevant stakeholders to monitor in real-time the currently ongoing personalization activities. The availability of such monitoring tools can be especially important in field trials, where incorrectly specified rules can lead to users misinterpreting the behaviour of the system [8] or, in the worst case, to unexpected and even dangerous situations [21, 35, 43]. Until now, several approaches have considered monitoring users' behaviour but without analysing the associated automations. One example [39] focused on using sensing technology to monitor a broader scope of so-called `meaningful activities', i.e. those satisfying person's emotional, creative, intellectual, and spiritual needs. The authors developed a toolkit of off-the-shelf, affordable sensors to allow persons with dementia and Parkinson's disease to monitor their meaningful activities. They describe two evaluations of the toolkit, firstly a lab-based study to test the installation of the system, and secondly an in-the-wild study in which the feasibility of the toolkit to monitor activities in and around real homes was evaluated. However, the authors involved subjects who were not target users of the toolkit, but were identified as 'technology enthusiasts'. Another study [20] reports on how 12 households with DIY smart home systems performed for two years, and studied the participants' strategies for maintaining their system awareness, but they did not provide any particular tool to support analysis of personalization activities.

Interest in the possibility of better understanding the personalization activities expressed in terms of interaction with the EUD tools to understand the personalization preferences of users has been limited so far. The attempts that can be mentioned in this regard mainly involve the analysis of repositories of IFTTT rules, which have been done for various goals. For instance, a study [6] shows how IFTTT can be susceptible to attacks by malicious applet makers, including stealth privacy attacks to get private photos, leak user location, and eavesdrop on user input to voice-controlled assistants. To this aim, the authors used a dataset of 279,828 IFTTT applets from more than 400 services, and they classify the applets according to the sensitivity of their sources, finding that 30% of the applets may violate privacy. Another example is the work of Ur et al. [42], who analysed more than IFTTT 200,000 applets created by over 100,000 users by considering the connections between the triggers and actions within the existing applets. However, these contributions only analysed external repositories of rules to identify relevant patterns and derive users' personalization preferences offline. Such work neither provided users with usable visual analytics means for deriving key information from such analysis, nor provided tools for supporting such monitoring in real-time.

Regarding visual analytics for the IoT domain and the need for providing usable representations of IoT-based data, a review of work on IoT and big data [33] discusses their usefulness in creating effective applications and services for various domains. An approach to repurposing Web analytics for the IoT is presented in [30], to add analytics to IoT deployments with minimal effort and cost. Previous work [27] highlights that with the growing adoption of smart home technologies, inhabitants have the challenge of making sense of the data that their homes can collect to configure automated behaviours that benefit their routines. However, current commercial smart home interfaces usually provide information on individual devices instead of a more comprehensive overview of a home's behaviour. In general, all such previous work did not consider the analysis of the personalized automations obtained by interacting with EUD tools. In this regard, an initial attempt to provide a visual analytics tool for analysing rule-based personalized behaviour created by users was presented in [11]. However, that tool was only able to provide information associated with one single user per time and just provided information on created rules but not on the rules actually executed. In contrast to all such previous work, in our approach, we not only provide usable visual representations for allowing relevant stakeholders to make sense of gathered data, but we also provide a fully integrated working tool to analyse also those rules that have been actually activated and triggered, thus providing useful insights on their actual use.

One previous contribution that focused on the execution of rules [29] reported on an empirical study performed on IFTTT ecosystems by collecting data for 6 months and profiling how data were used, as well as the performance of applets (IFTTT rules) execution. The analysis of the collected data shows that the services provided by the IFTTT ecosystem involve mostly smart home devices (e.g. lights, thermostats, Amazon echo, etc.) and hubs (e.g. Samsung Smart Things), and such services may act both as triggers and as actions. However, that work just reports an empirical ad-hoc study for analysing the IFTTT ecosystem using a custom testbed in controlled experiments, therefore it does not provide any tool that can be used to analyse different case studies. To sum up, to our knowledge, there is a lack of solutions able to support execution and analytics of personalization rules in real daily environments equipped with IoT technologies, a lack which we aim to fill with this work.

4 Background

In this section we provide background information to better introduce the context and the scope of our current research, by briefly summarising some relevant contributions we already put forward in the past and their main motivations.

We presented an initial proposal [17] for the design and the development of an automatic environment for specifying personalization activities in IoT scenarios through trigger-action rules. That environment just allowed end users to create, modify, save, delete and reuse trigger-action rules (e.g. import them from a shared repository. That tool was assessed through a usability study to gather some relevant user feedback. In that contribution, an initial ideation of a high-level general solution architecture of a platform supporting personalisation through trigger-action rules was also introduced. It is worth noting that such initial architecture, over the years, significantly evolved from a first ideation to a concretely working solution supporting the actual execution of the actions contained in rules, and even providing 'smarter' support in some specific cases (e.g. reverting back automatically the effect of some rules, without requiring the user to explicitly specify them in additional rules, see Section 5.1 for further details). According to the evolution of the platform, also the included environment for managing rules significantly improved. For instance, the "Rule Editor" presented in [17] only supported the editing of rules, and it was only integrated with a context manager middleware, whereas the current "Tailoring Environment" is fully integrated with relevant components of the platform, and also allow users to send rules for their actual execution in real settings (by activating or deactivating them). Section 5.1 will provide more details about the current functionalities supported by the Tailoring Environment.

However, despite some promising results derived from usability studies involving EUD tools, it became soon clear that, even providing users with usable EUD tools, without proper support, they may easily define rules triggering actions that are in conflict, or not resulting in the intended behaviour. Therefore, we focused our attention on developing a solution supporting end-user *debugging* exploiting the *simulation* of the context in which trigger-action rules are expected to be executed [26]. In particular, the tool presented in [26] supported functionalities not only for simulating rules but also for 'debugging' them (i.e. find possible errors and provide indications to fix them) by providing *conflict resolution support* (for identifying possible conflicts within a set of rules) as well as *interactive explanations* (i.e. the system answers relevant questions), so that users can better understand the source of their errors. It is worth noting that the system was yet not able to support the actual execution of rules in real environments.

As soon as users exploit EUD tools in their common practise, they increasingly add new rules or modify/delete existing ones: therefore, the list of rules currently saved in their repositories can be a key source of information to understand their personalisation needs. Thus, another contribution in the direction of better supporting end users in specifying their personalization rules was the development of a visual tool [11] for better analysing how users interact with a EUD tool. This tool allows understanding not only end-users' personalization needs and how users express them, but also whether users encounter difficulties in interacting with the EUD environment itself. However, that system was not able to show any information about the characteristics of the rules that were actually *executed* in the user's environment but it provided information only on the rules created/modified. In addition, it only provides information associated with one single user per time.

Finally, in [24], considering the increasing availability of humanoid robots in various domains of everyday life, we focused our attention on how to enable those who are not programming experts to personalize robot behaviour according to contextual information detected by both the robot and by the available IoT objects and sensors available in a specific context and to link the robot behaviour to what happens around it so as to describe the robot behaviour in a context-dependent manner. The solution was assessed through an in-lab usability study in which users had not only to specify relevant rules involving the Pepper robot, but they also had to execute them –though only in the lab– to offer to involved participants the possibility to see the effect of their interactions with the EUD tool in terms of actual execution of rules.

In the current work, we provide an original contribution connected with two main aspects that were not touched by such previous work. First, the support offered by our platform to execute the personalization rules "in the wild", also providing our experience with real users in their contexts to show the feasibility of the approach. Secondly, we propose the possibility of visualising and analysing the rules that are actually executed in the real users' environments.

5 The Platform

Considering previous experiences in the field [17], we have designed and built a new system for specifying, executing, and managing personalization activities in IoT scenarios through trigger-action rules, with particular attention to introducing monitoring features in the platform, to support a better understanding of its actual use in the wild. Figure 1 shows the architecture of the considered software platform. It includes a Tailoring Environment through which even people without programming background (e.g. domain experts, end users) can specify the desired personalization rules. The Tailoring Environment sends such rules to a Rule Manager, which receives information from the Context Manager when the triggers involved in the rules are verified: when this happens, the Rule Manager sends the actions to execute to relevant applications and appliances. The Context Manager is a software composed of one server and various delegates. The purpose of the Context Delegates is to communicate directly with the various sensors or other entities able to generate events, to be informed when the state of their associated variables changes. When this happens, the Context Delegates communicate such changes to the Context Server, according to a pre-defined vocabulary used to specify the triggers. There is also a Monitoring module which aims to provide support for analysing the use of the personalization platform, by showing relevant information to users interested in it (e.g. in Ambient Assisted Living scenarios they can be caregivers or platform managers). For gathering the relevant information, the Monitoring module receives data from the Rule Manager concerning the rules that have been executed. This module is a key added value because it allows relevant stakeholders to focus on the personalization that has actually been put in place by users, thereby of actual interest for them. Applications can be integrated with the platform in order to: receive actions (which were included in relevant rules) indicating requests of modifications to make, and/or send events generated by the application itself (in this case the application acts as a Context Delegate, by sending the information associated with the occurred event(s) to the Context Server).

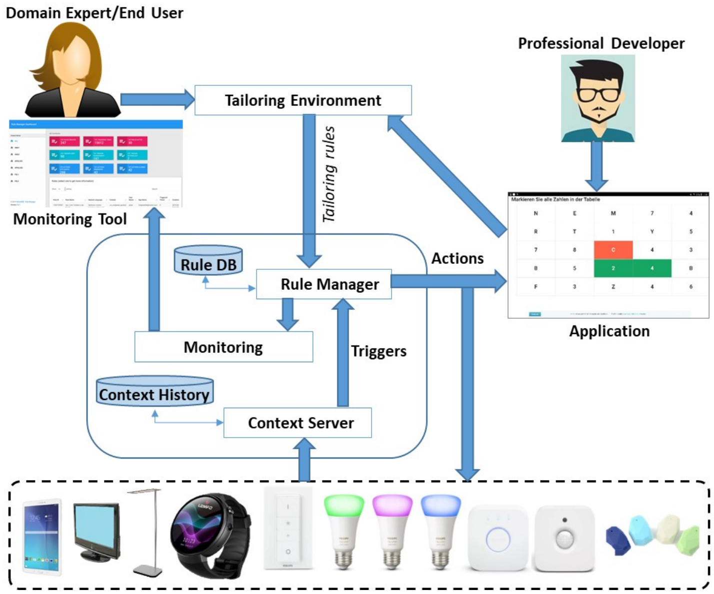


Figure 1: An Overview of the Main Components of the Platform and Their Communications

Another novel feature of the platform is the support it provides when there are rules that in the trigger part have a condition and in the action part have an action that involves a change of state of an actuator (the so-called "sustained" actions [21]), and this state can assume only two opposing values (e.g. "true"/"false", or "on"/"off"). An example of such kind of actions is turning on/off the lights for instance: "*IF time is between 20 and 23:30, DO turn on the living room light"*. Its trigger part is a condition, whereas in its action part there is a sustained action which affects the actuator (light) state, which can assume only two possible values (on or off). However, two different interpretations could be given to the possible effects this rule can bring about:

(i) The living room light will be ON between 20:00 and 23:30;
(ii) The living room light will be ON between 20:00 and 23:30 AND the living room light will be OFF otherwise (i.e. when the time is before 20:00 OR the time is after 23:30)

The first behaviour (i) just turns on the living room light in the defined period, leaving unspecified the behaviour of the system in the remaining intervals of time (namely: before 20:00 and after 23:30). Instead, the second behaviour (ii) not only keeps switched on the living room light between 20:00 and 23:30, but it also automatically reverts that action as soon as the condition is no more verified. This means that it automatically switches off the light when the time does not belong anymore to the specified time interval. When asked about such two possible interpretations, users declared that the second behaviour is the one that people would typically expect from a smart system. Thus, the second behaviour is the one that is now supported by the platform. It is worth pointing out that both (i) and (ii) are different from the case when the trigger is defined as an *event* (e.g. "*WHEN the time becomes 20:00, DO switch on the living room light*"): in this case, the action is executed when the time *becomes* 20:00, but nothing will be automatically carried out by the system to revert that action (i.e. switch *off* the light at some point).

5.1 The Tailoring Environment

The Tailoring Environment is the EUD environment through which users specify personalization rules. On the one hand, it shows relevant triggers, which describe the main aspects that may change in the context, categorized under a hierarchical representation having three main dimensions (*Users*, *Environments*, *Technology*) at its highest level. On the other hand, it also provides users with a structured representation of actions, which mainly depend on the specific application(s) considered. The Tailoring Environment is designed to be a generic environment which can be easily configurable, making its customization to support specific applications, domains and contexts an easy task. Figure 2 shows the panel that allows for configuring the key parameters to instantiate the platform for a specific case:

• *"User id"* specifies the name of the user who is currently using the Tailoring Environment. This information is important to correctly associate the rules created through the Tailoring Environment with the correct user;
• *"Context Manager URL"* specifies the URL of the specific instance of the Context Server used. This information is essential to identify the triggers that the Tailoring Environment has to show, which depend on the considered context;
• *"Rule Manager URL"* specifies the URL of the Rule Manager. This information is key to understand the URL to which to send the verified rules for their execution.
• *"Application Set"* specifies the name of the application(s) considered. This information is important to identify the actions that can be triggered.

To properly support the specific, actual context of use at hand (e.g. a specific home), at the beginning the Tailoring Environment needs to be 'populated' through the triggers and the actions actually available in the considered context. To this aim, at initialization time, on the one hand, the Tailoring Environment asks the instance of the Context Manager specified in the "*Context Manager URL"* previously mentioned) the description of the current context of use, in order to populate the panel dedicated to triggers. The Context Server will then provide such a description in JSON format. In this way, if a new smart object (or a device) is registered in the Context Server (through a new Context Delegate), the Tailoring Tool will receive from the Context Server the updated description of the triggers currently available in the considered context. On the other hand, still at initialization time, the Tailoring Environment also asks the concerned application(s) about the specific actions to make available in the user interface part of the tool dedicated to actions.

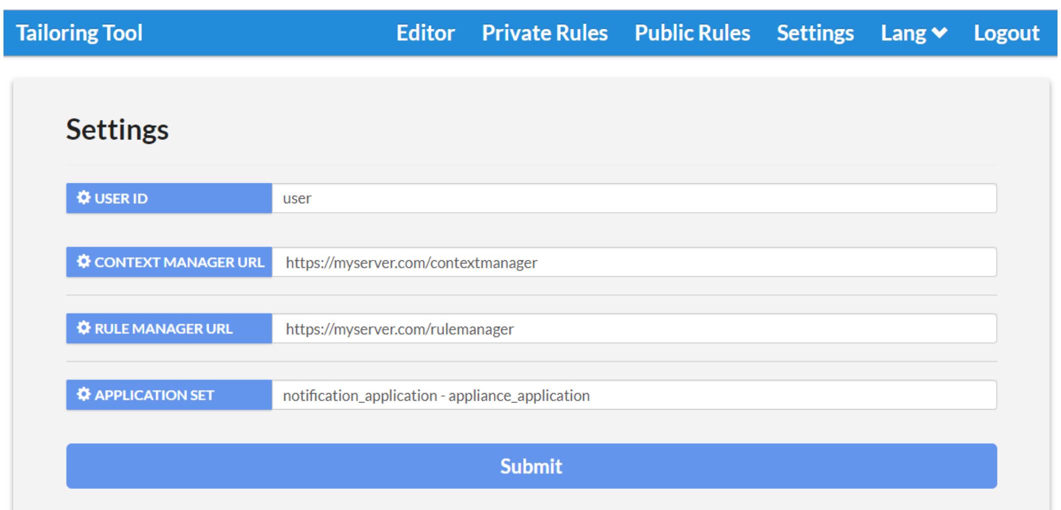


Figure 2: The "Settings" Panel of the Tailoring Environment

In the top part of Figure 2 there is the menu to access the different parts of the Tailoring Environment. In particular:
• "*Editor*": the selection of this element shows the user interface of the Tailoring Environment, which visualises the hierarchies of triggers and actions;
• "*Private Rules*": by selecting this item, the list of rules that users have in their private repositories will be shown. After creating or modifying rules, users can save them only in their private repositories;

• “*Public Rules*”: by selecting this item, the list of rules that users share in a public repository will be shown. Users can import rules from the public repository, or they can export rules from their private repository to the shared one;
• “*Settings*”: by selecting this item the panel shown in Figure 2 will be visualised;
• “*Lang*”: it activates a pull-down menu for selecting a specific language (currently, English, Italian, Romanian, Norwegian and German are supported);
• “*Logout*” allows the user to exit the tool.

The rule language used for the EUD tool specifies the contextual events and conditions to consider and the consequent changes to be executed accordingly. Rules are specified according to an ECA-based (Event, Condition, Action) format specified through an XML Schema Definition (XSD). *Events* are associated with a change of the state of a contextual entity. They can be either elementary or complex events, namely those obtained by composing events through Boolean, comparison or sequential operators. *Conditions* refer to a state of a contextual entity. Conditions can also be either elementary (e.g. a Boolean predicate) or complex (i.e. obtained by composing elementary conditions). *Actions* specify the changes that should be applied to e.g. a smart object, a device or an application. They are typically aimed at changing appliances' state, or at activating some functionalities. The “action” part of a rule can specify a single action or a set of actions, to apply in a sequential or parallel manner.

The Tailoring Environment also supports the explicit distinction between events and conditions (e.g. using the dialog visualised in Figure 3). This distinction was introduced considering previous user studies [21], which have shown that EUD environments sometimes create confusion in end users concerning the difference between these two concepts. This issue was also confirmed in our direct experience when we started to introduce the tool to our users. Thus, by introducing the explicit distinction, we better stimulate users in thinking about their difference. In addition, the top part of the central panel of the Tailoring Environment provides feedback in natural language of the created rule, to express the specified behaviour in a more immediately understandable manner.

Moreover, it is possible to define rules that are triggered if an event does *not* occur in a specific interval of time (e.g. *When the medicine has not been taken between 10 a.m. and 11 a.m.*), which were found useful in some scenarios. This has been supported thanks to the introduction of the NOT operator associated with an event in the proposed tool, differently from other solutions that do not support it (e.g. IFTTT). Another useful feature of the rule language used is the possibility to define rules that are triggered when a specific ordered sequence of events occurs (e.g. “*If the user enters inside a room and then he exits*”; or “*When the temperature becomes more than 20 degrees and then the humidity level becomes more than 50%*”), or when an event occurs a specified number of times (event iteration), e.g. “*If the user goes to the bathroom 5 times during the night*”.

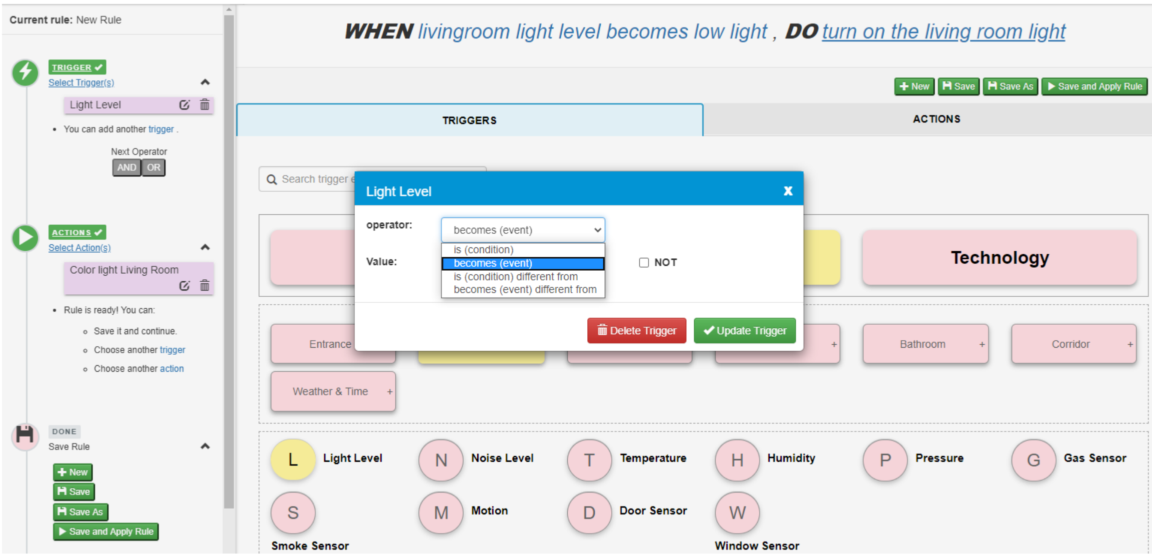


Figure 3: Distinguishing between Events and Conditions in the Tailoring Environment

As we mentioned before, the rules created are saved in a private repository (an example is in Figure 4), where some pieces of information are provided for each rule: priority, name, description in natural language, the last time when the rule was modified. In addition, associated with each rule, there is also a "Edit" button to change parts of an existing rule. By selecting one or more rules in the tool (using the associated check-box shown in the left-most column of the panel shown in Figure 4), and then selecting the "Activate Rule" button, the user can make 'active' the selected rule(s). This means that they are sent to the Rule Manager for enabling their possible execution: they will be executed only when the included triggers are verified. Once a rule becomes *active*, which means enabled for execution, a green background highlights it (see Figure 4). Whenever users want to de-activate one or more rules, they have to select such rule(s) and then select the "De-activate Rule" button. This action will remove the selected rule(s) from the list of active rules in the Rule Manager, and consequently, the rule will no longer be considered for its possible execution (until it is activated again). Moreover, it is worth pointing out that, by using the Tailoring Environment, it is currently possible to activate some rules also on behalf of other users: the users that have this possibility can be selected among a 'permission' list available within the tool.

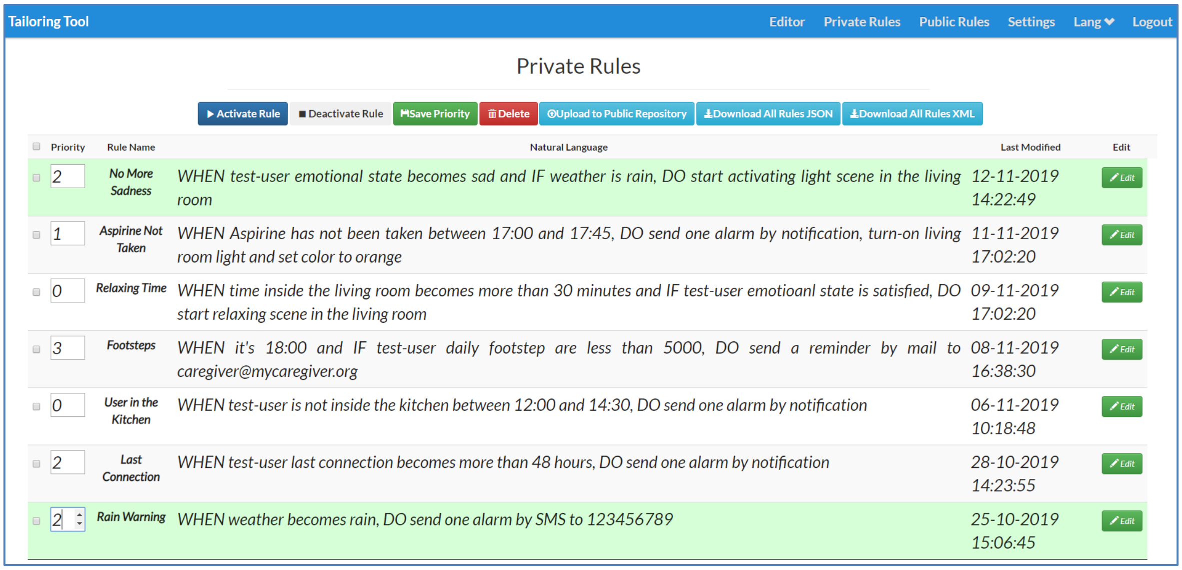


Figure 4: Visualizing the List of Rules in the Tailoring Tool

5.2 The Context Manager

The main task of the Context Manager is to maintain an up-to-date picture of the current situation of the considered context of use (e.g. the house of a specific user), and to inform the other modules of the platform -the Rule Manager in the first place- when relevant updates to such a context occur, i.e. when an event happens or a condition is fulfilled. Therefore, the Context Server communicates with the Rule Manager only when the triggers specified in a rule are verified.
The Context Manager is a distributed module, composed of a Context Server and some Context Delegates. The Context Delegates are software components that communicate with the sensors and the appliances available in the considered context: they get raw information and send it, in a suitable format, to the Context Server, by using a RESTful service that this module exposes to this aim. In turn, the Context Server, according to the data received from the Context Delegates, updates a database storing the current and past values (historical data) of the attributes that the various elements considered in the concerned context dynamically assume over time, and represented using a common format. More specifically, the description of the types that such contextual entities can assume, and the hierarchical structure in which they are organised are represented in a context meta-model specified in an XSD format. When the Context Server is compiled, this XSD file is automatically translated into a set of Java classes, and various instances of Java objects are created to define the state of the elements composing the current context (e.g. the various instances of users, environment and technologies). In order to update such Java objects, the Context Server provides just one RESTful service to receive the data from the various Context Delegates which, using this REST service will be able to update the various attributes of the context. The input parameters of this service aimed to update the information received from the Context Delegates are: i) the *id* of the corresponding context dimension; ii) the *xPath* value indicating where the context attribute to update is located within the context model structure; iii) the *new value* gathered from the sensor.

For example, the trigger representing the event "user has fallen" is defined in the context model as an attribute of the "User" dimension called "layingDown", and it assumes a Boolean value: this trigger is defined under the following hierarchy: "user -> physical -> laying down". To update the *layingDown* attribute, the context delegate sends three parameters to the REST service: the user id, the xPath of the attribute to update (in this case "user / physical / @ layingDown") and the current value of the attribute (in this case "true"). The advantage of this single-service solution is in terms of platform evolvability. Indeed, when there is the need of updating the context meta-model (because of a new context element type, or an existing context element type has been changed, or removed), it is sufficient to modify the XSD definition of the context metamodel, since the REST service above mentioned can still be used without requiring any further changes. This is because: i) the JAVA classes that are expected to manage the values of the attributes of the updated context are automatically generated when compiling the Context Server; ii) the above-mentioned RESTful service will still be able to support changing the values received for the newly generated context element(s). For instance, suppose that there is the need to introduce a new attribute providing information about user movements (i.e. "isMoving"), and this attribute needs to be added under the "user/physical" xPath hierarchy (namely: "user / physical / @isMoving"). In this case, while the XSD describing the structure of the context needs to be modified, there is no need to create a new, ad-hoc service to receive and update values of this newly introduced attribute, because the RESTful service described before can still be used to change the value of the concerned "isMoving" attribute. This is realised through the JAVA reflection mechanism (namely: the capability of an executing Java program to examine upon itself, i.e. obtain the names of all the fields or methods of a JAVA class), which makes it possible to automatically derive, within a Java program, the name of the method to call (by using the information included in the parameters mentioned above), and consequently update it. This type of solution is modular and evolvable because it can easily manage flexible introduction of various sets of sensors and applications, according to the specific needs to address, in a dynamic manner.

5.3 The Rule Manager

The Rule Manager has a twofold functionality. The first one is to receive the rules from the Tailoring Environment and store them in a database. The second one is to act as an in-between module between the Context Manager and the applications. On the one hand, the applications subscribe to the Rule Manager to receive the actions that describe the behaviour expected when a rule is triggered. On the other hand, for each application, the Rule Manager retrieves the rules associated with that application, and subscribes to the Context Manager for being notified whenever the triggers included in such rules are verified. The MQTT protocol has been used to support the communication between the Rule Manager and the application, which in turn communicates with the concerned controllers, gateways or applications to send the actions that should be executed. As soon as a user of the Tailoring Environment activates a set of rules, such rules are sent out to the Rule Manager, which stores them in a database. These rules are the so-called 'active' rules. i.e. those that are enabled for their *possible* execution (they will actually be carried out when the involved triggers are verified). Depending on the triggers involved in these active rules, the Rule Manager subscribes to the Context Server for being notified whenever the triggers contained in the active rules are verified. As soon as one or more of such triggers are verified, the Rule Manager receives a notification from the Context Server and then selects accordingly the corresponding actions to execute, which are sent out to the relevant application for being executed. The Rule Manager also sends to the Monitoring module the information about the time when such actions have been triggered.

5.4 The Monitoring Tool

The main goal of this module is to provide its users with information about what happens in the end-user context (e.g. seniors' homes), both in terms of activities done by the users (as detected by sensors), and in terms of personalization activities that have been put in place through the specified rules. To this aim, this module receives input from the Rule Manager : it receives information about personalization rules that have been sent for execution and about the time when they have been actually triggered.

The information about triggered rules can represent valuable data to understand what is currently and actually going on in one or more end user sites, to identify the personalization aspects which users are focusing on most, the types of routines they have put in place and the frequency with which such automations occur. Indeed, users of the Tailoring Environment could create/add several rules in their repositories over time.

Thus, it can happen that some rules appear in the repository, but they are not currently exploited e.g. the user did not want to have that personalization active currently. Instead, the information about the triggered rules shows evidence of rules that have been in the past (or are still currently) of actual interest for their users. Other information that can be obtained by exploiting such a tool is the one associated with how many times a specific rule has been triggered. This value is highly dependent on the purpose of each particular rule.

In the following, we will better detail the information provided by the Monitoring Tool, which provides various types of structured information. Therefore, to facilitate its description, in Figure 5 we schematized its layout as structured into four main parts. The content of such four parts are further detailed in the following four sections (from Section 5.4.1 to Section 5.4.4).

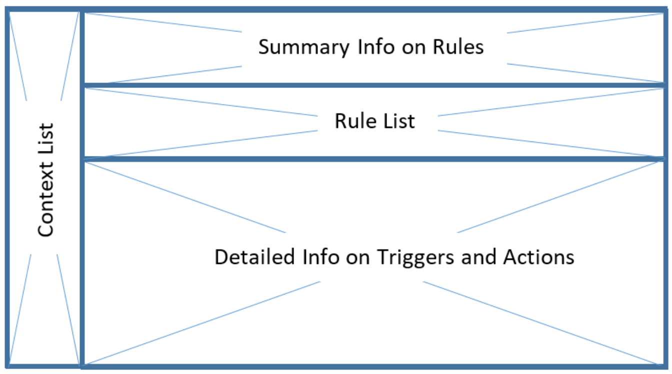


Figure 5: The layout structure of the Rule Monitoring tool

5.4.1 Context List

When the Monitoring tool is accessed, it shows the list of the contexts associated with the various platform instances deployed in the considered contexts (see the vertical left-hand panel shown in Figure 6). In addition, there is a further option ("ALL", see Figure 6), which provides an overview of the information associated with all such contexts. When a specific context is selected in the left-hand panel, the information about such context is shown in the central panel of the tool.

5.4.2 Summary Information on Rules

This part (see the main panel visualised in Figure 6) shows various pieces of information associated with all the contexts or just one context, depending on what is selected in the Context List. Starting from the top, there is a first row showing three pieces of information about rules:

(iii) *Rules created*: the total number of rules created by the considered user(s);
(iv) *Triggered times*: how many times the rules have been triggered;
(v) *Rules active*: the number of rules that are currently active and therefore could be executed (they will be executed as soon as the involved triggers are verified).

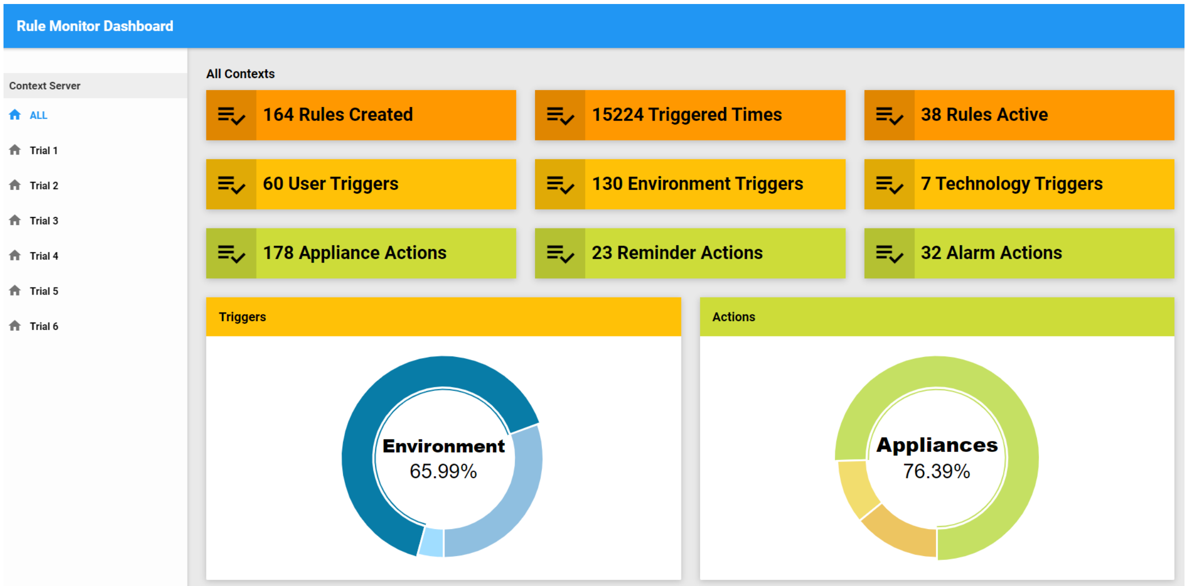


Figure 6: Visualizing summary information on rules

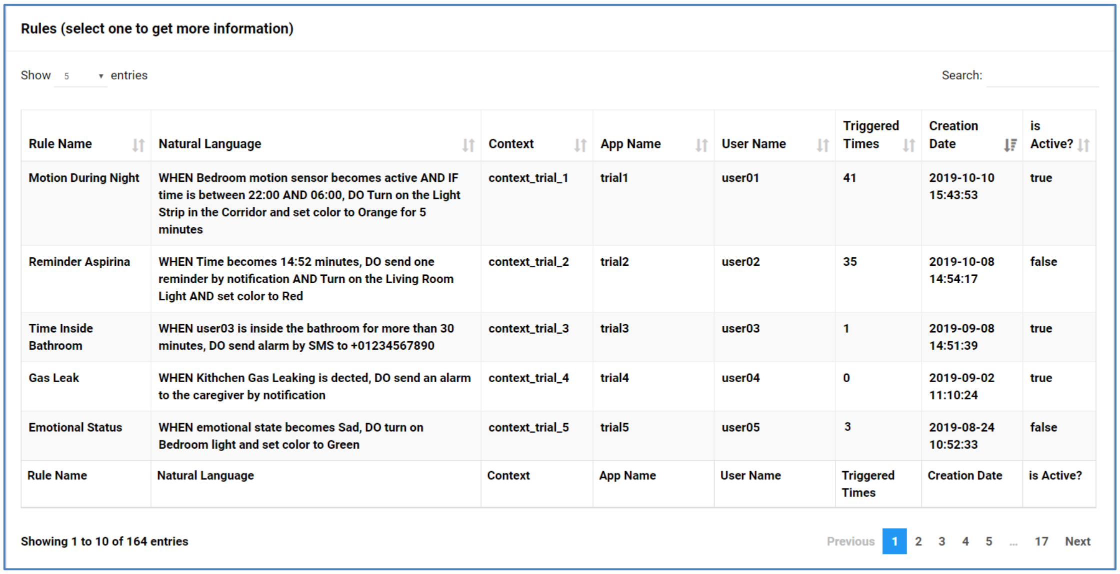
The second row shows information about the triggers involved in the created rules, grouped according to the three main contextual dimensions (*User*, *Environment*, *Technology*). The last row displays information about the actions appearing in the created rules, grouped by their type (appliances, reminders and alarms).

Besides such summary quantitative indicators, there are also two interactive pie charts (see the bottom part of Figure 6) visually presenting how the various types of triggers and actions are distributed. In particular, the first pie chart shows, for each contextual dimension (which can be interactively selected by clicking on the corresponding portion of the pie chart), the percentage of triggers of that type used in the created rules. Similar information is provided in the second pie chart, as associated with actions. The percentage of each type of action used in the created rules is shown as a portion of the pie chart: when users interactively select one portion, further (textual and numerical) information is displayed accordingly. In the example in Figure 6, the "Environment" category was the most used contextual category, whereas the "Appliances" category was the most used action type.

5.4.3 Rule List

In this part (which corresponds to the "Rule List" portion in Figure 5), a detailed description of all the rules created is shown. In particular, for each rule the following information is provided (see Figure 7):

- *Rule Name*: the name of the rule, as given by its creator;
- *Natural Language*: the specification of the rule in natural language;
- *Context*: the context to which the rule refers;
- *App Name*: the application to which the rule refers;
- *User Name*: the creator/owner of the rule;
- *Triggered Times*: how many times the rule has been triggered, in total;
- *Creation Date*: when the rule has been created;
- *isActive?*: whether the rule is currently 'active' or not. "Active" means that a rule is currently included in the set of rules that the Rule Manager considers for possible execution.



Rules (select one to get more information)

Show 5 entries Search:

Rule Name	Natural Language	Context	App Name	User Name	Triggered Times	Creation Date	is Active?
Motion During Night	WHEN Bedroom motion sensor becomes active AND IF time is between 22:00 AND 06:00, DO Turn on the Light Strip in the Corridor and set color to Orange for 5 minutes	context_trial_1	trial1	user01	41	2019-10-10 15:43:53	true
Reminder Aspirina	WHEN Time becomes 14:52 minutes, DO send one reminder by notification AND Turn on the Living Room Light AND set color to Red	context_trial_2	trial2	user02	35	2019-10-08 14:54:17	false
Time Inside Bathroom	WHEN user03 is inside the bathroom for more than 30 minutes, DO send alarm by SMS to +01234567890	context_trial_3	trial3	user03	1	2019-09-08 14:51:39	true
Gas Leak	WHEN Kithchen Gas Leaking is dected, DO send an alarm to the caregiver by notification	context_trial_4	trial4	user04	0	2019-09-02 11:10:24	true
Emotional Status	WHEN emotional state becomes Sad, DO turn on Bedroom light and set color to Green	context_trial_5	trial5	user05	3	2019-08-24 10:52:33	false
Rule Name	Natural Language	Context	App Name	User Name	Triggered Times	Creation Date	is Active?

Showing 1 to 10 of 164 entries Previous 1 2 3 4 5 ... 17 Next

Figure 7: Showing Information about the Rules Executed

For instance, in the rule list shown in Figure 7, it is possible to see that the rule named "Emotional Status" has been triggered three times in the past, whereas currently it is not active. This list of rules is also interactive: when users select a specific rule, they can obtain further information about it, in particular on the events, the conditions, and the actions involved.

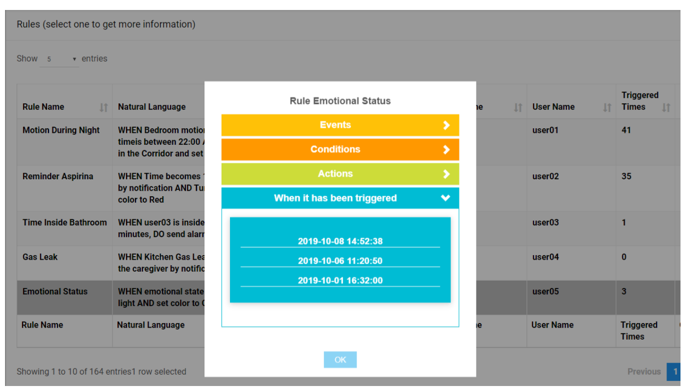


Figure 8: Showing information about the time when a rule has been triggered

In addition, when a specific rule has been triggered at least once, users can get further details about when each execution occurred (see Figure 8), by double-click on that rule.
Finally, it is possible to order the visualisation of such rule list according to various included fields (e.g. according to the creation time).

5.4.4 Detailed Info on Triggers and Actions

Detailed information about the context dimensions involved in the created rules (in terms of event types and condition types), and about the involved actions is shown in the dashboard visualised in Figure 9. This information is useful to understand which triggers and actions have been exploited by users for creating their rules. In particular, the dashboard shows three interactive pie charts visualising the frequency of use of the types of triggers and actions involved. By selecting a specific portion of the pie chart, the user gets more precise information (in percentage terms) about the frequency with which the associated event/condition/action type occurs in the created rules. For instance, Figure 9 shows that "Bedroom" is the element of type "Environment" that was used, as an event, in about 19% of total cases, whereas "Weather-time" was used as a condition in around 41% of rules. For the actions, the "Notification" type was used in about 9% of cases.

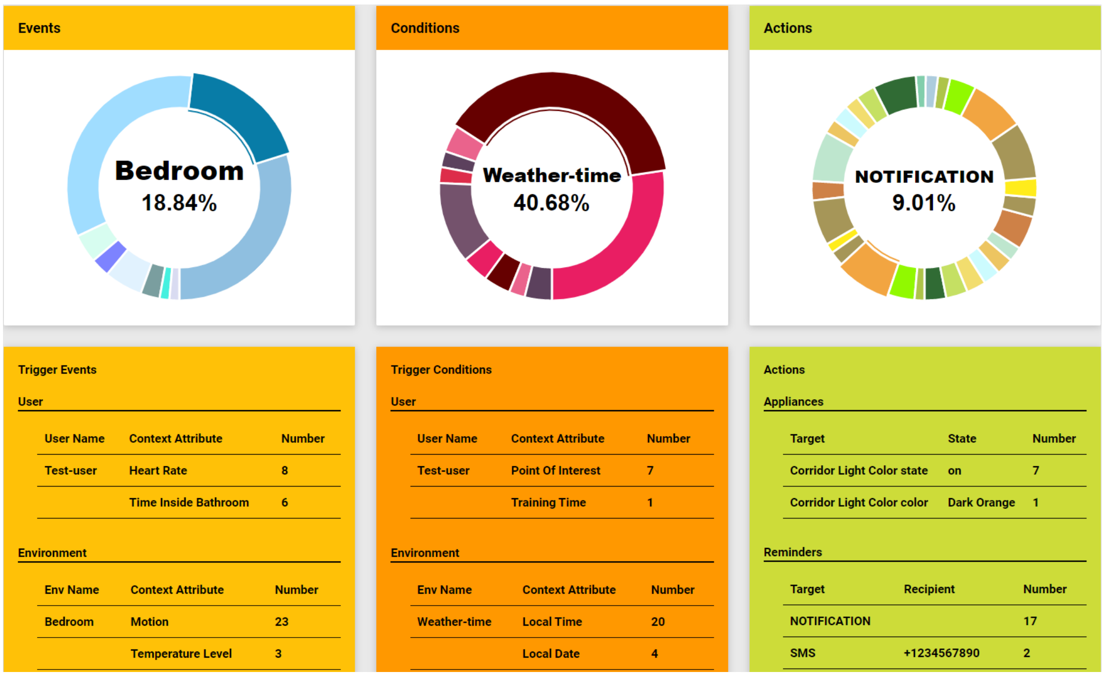


Figure 9: Showing Detailed Information about Triggers and Actions

While such pie charts visualise the event/condition/action occurrences only in percentage terms, more precise data about such occurrences are shown by the tool just below such pie charts, through some textual tables (see the bottom part of Figure 9). To better explain the meaning of the information contained in such tables, we will focus on those dedicated to "Trigger Events" (i.e. triggers of 'event' type, which have been involved in the created rules), visualised in the textual tables shown in the left-most bottom part of Figure 9. In such tables, the information is categorised according to the various trigger dimensions, by providing a separate table for each dimension. However, in Figure 9, for the sake of legibility, some information provided in the bottom part has been omitted (i.e. only two categories, User and Environment, are currently visualised for "Trigger Events", thus neglecting "Technology"). The table dedicated to the "User" dimension presents the information according to three different fields: "User Name" (the name of the considered user), "Context Attribute" (the name of the specific contextual trigger considered, under the "User" dimension) and "Number" (the number of times the concerned trigger appears in the created rules). In particular, the contextual attribute "Heart rate" of the user named "Test-user" was involved 8 times within the created rules, whereas the trigger named "Time inside Bathroom" was involved 6 times. By analysing the 'Environment' category, it is possible to see that the "Motion" trigger associated with "Bedroom" environment appeared 23 times in the created rules, whereas in 3 cases the rules involved the detection of the temperature in the bedroom. Thus, the information associated with "Bedroom" in the textual table gives a more precise explanation of the percentage visualised in the corresponding pie-chart (where it is shown that "Bedroom" appeared in around 19% of rules). Along the same line, similar information is shown for the triggers of condition type, and the actions.

6 Deployment

6.1 Platform Configuration

In order to investigate the suitability of the approach in a real case study, we deployed an instance of the platform in the context of an AAL project, which considers how to better support elderly in their daily activities (mainly at home) as well as their formal or informal caregivers.

The first issue was to decide the set of sensors and appliances to use in the platform deployment. For this purpose, we considered components available on the market and having a reasonable cost. In the end, the selection considered several lighting components and various types of sensors and devices. For the lighting components, the choice was driven by the possibility of conveying specific effects to better support older adults: to direct their attention promptly (because light attracts attention); to improve their sleep-wake-rhythm (by using e.g. different colour temperatures and illuminance levels at different times of the day) because often the elderly suffer from sleep-related disorders; to support the structure of their daily activities (e.g. remind specific activities); and to help seniors finding the right place easier (e.g. finding the bathroom during the night).
To this aim, we used various off-the-shelf lighting components, and a standing luminaire (developed in an external project) having the possibility to activate a biodynamic light curve varying in colour temperature and light intensity in a 24-hour rhythm, and to create activating or calming room ambiences with different light scenes. The luminaire was placed in the living room, as it typically is one of the most used rooms in a house. In addition to the standing luminaire, various types of Philips Hue lights were exploited to support similar effects in other parts of the house. In particular, we used a Philips Hue Light Strip (for illuminating the bedroom-bathroom path), Philips Hue Dimmer Switches (to control the light in various rooms of the house), and a Philips Hue Go (a portable and remotely controllable light).

For indoor location, we adopted a solution based on the use of a smartwatch in combination with the support of some Estimote Proximity Beacons[2] , to obtain easy-to-install support. We discarded solutions based on smartphones since they may be easily forgotten in the home (distant from where the user actually is), while the user moves across the house. Solutions based on cameras were discarded for privacy reasons. The selection criteria for the smartwatch included the capability to support at the same time both the communication through Bluetooth (to detect the beacons placed in the home) and through Wi-Fi (to communicate the smartwatch's position to the platform). However, unfortunately, most smartwatches currently on the market do not support this feature. Thus, we used the Lemfo LEM7 smartwatch, which can connect and communicate at the same time through Bluetooth and Wi-Fi. The smartwatch also provides other information about the user, such as heart rate and steps.

The platform configuration also includes an Android-based tablet (Samsung Galaxy A SM-T580), on which a cognitive stimulation and a Reminder/Alarm application are running. The cognitive stimulation application was connected to the platform to use the results obtained from playing with its game exercises (or other information provided by them) as possible rule *triggers* (within rules such as "When the result of the exercise becomes not satisfactory, do send a message to the caregiver"). The triggers related to the game are "Emotional State", "Cognitive State", "Training Result", "Training Time" (how much time the user trained with the application) and "Time since last connection" (how many hours passed from the last time the user used the application).

Moreover, a further set of sensors is used to detect surrounding user's contextual conditions, namely: motion, illuminance level and temperature (provided by Philips Hue Sensor), gas (Honeywell Gas detector), smoke (Honeywell Smoke detector), humidity/temperature/pressure (Xiaomi sensor), use of pill dispensers (Minew

[2] https://estimote.com/

E8 Beacon), and whether windows or doors are open (Xiaomi Mijia Door/Window Sensor). Furthermore, the configuration also included further equipment mainly serving as hubs, i.e. they collect data from sensors. An IoT gateway (the Geniatech GTW410 Gateway) was used to support the connection between the different hubs and the platform

Figure 10 shows how the platform was deployed, the communications between the different architectural modules, and (a part of) the used sensors and objects, also specifying further elements that were needed to deploy the platform in the real trial contexts considered. In particular, rules are created by the domain expert through the Tailoring Environment (1), which sends them to the Rule Manager that stores them (2), and it also subscribes to the Context Server (2') to be informed when a rule is verified. When events occur in the concerned context (3), they are used to update the Context Server, which in turn notifies the Rule Manager as soon as triggers belonging to rules are verified (4). When this happens, the Rule Manager sends the corresponding actions (5-6), which can involve an application or an appliance. The first case is represented by the situation when the action is an alarm or reminder rendered through a notification on the tablet where the application is running (in this case, the actions are sent to the application via MQTT). When appliances are involved in the actions (second case), the actions coming from the Rule Manager are received by an OpenHab[3] (OH) Add-On we developed for this purpose through the MQTT channel (further information about it will be provided later on in this section). This add-on (which runs on the Geniatech gateway) is able to interpret such actions and, depending on the specific type of appliance involved, it sends the command to the associated gateway or bridge. For instance, the Philips HUE products (the bulbs, the LightStrip, the Portable Light and the Motion sensor), in order to be controlled, need a Philips HUE Bridge connected to the local network of the home (in our case, a router provided a Wi-Fi connection within the home, and the HUE Bridge was physically connected to the router via an Ethernet cable). Even the Luminaire is provided with a controller connected to the local network (through an Ethernet shield), and it is able to receive actions from the OpenHab Add-on by exposing RESTful services.

[3] https://www.openhab.org/

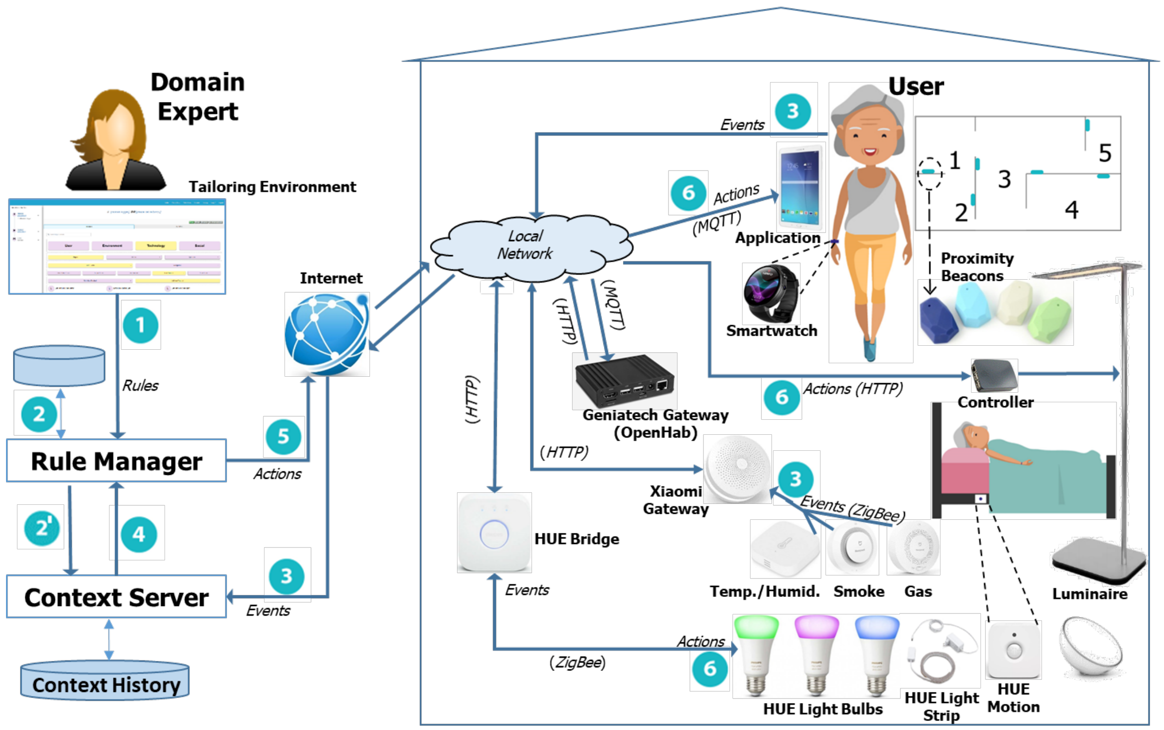


Figure 10: Platform Configuration used in Deployment

The connection to the platform of all sensors and appliances is done through an OpenHab installation running on a Geniatech Gateway GTW410, a Smart Home Gateway integrating ZigBee, Wi-Fi and BLE communication protocols to obtain remote data acquisition and remote-control functions. OpenHab is an open-source automation software that can support different home automation systems and technologies. To integrate OpenHab in the platform, we developed two OH add-ons. One OH Add-On subscribes to the platform at startup and waits for actions (which will be received through the MQTT protocol) coming from the Rule Manager. This add-on is also able to interpret the actions and apply them to change the state of the supported appliances (i.e. we implemented the support for the actions involving the Luminaire). Another OH Add-On acts as a Context Delegate: it monitors all the sensors supported by OpenHab (Xiaomi sensors for humidity, pressure, window/door open, smoke and gas detection; Minew sensors attached to medicine boxes; Philips sensors for motion, light and temperature) and sends the updated values to the Context Server.

6.2 Field Trials

Between May and September 2019, we carried out six field trials, which involved the deployment of the platform in six different apartments with older adult occupants located in two European countries (Romania and Italy). Both formal and informal caregivers and elderly patients were recruited during the requirements gathering, through focus groups and questionnaire administration.

Each field trial had the same test configuration (in terms of devices and sensors installed) described in the previous section, apart from some slight differences due to specific flat configurations and contingent technical issues. All volunteers participating in the field trials signed an informed consent form in which it was also specified that the person is entirely free to withdraw from the study at any time, at their convenience and with no penalty.

Before the trials, there was a preparation phase in which the platform was installed in each apartment. This phase was mainly done by the members of the project who represent end user organizations: one of such persons was assigned to each apartment. They were also provided with a guide explaining how to install the various sensors and smart objects in the house. In addition, in the days preceding the installation of the system in the flats, training about the concrete use of the system and its various components was also organized on-site for the older adults and the caregivers participating in the field trials. In particular, the training involved explaining the use of:

- the set of new lights with changing intensity and colour;
- the tablet, which displays notifications and supports playing with the games that helps the elderly in staying mentally active;
- the different sensors and devices (in particular the smartwatch);
- the actions that the system could potentially carry out automatically, e.g. notifications and changes in the lighting situation, as specified in the associated rules;
- the Tailoring Environment for creating rules personalised according to each older adult.

It is worth pointing out that, in the AAL domain considered, the primary users of the Tailoring Environment are supposed to be the (formal or informal) caregivers, who are supposed to have a good knowledge of the needs and preferences of their care receivers. However, the Tailoring Environment could be ideally used even by the elderly, if they are sufficiently accustomed to technology: this was not the case, because the technological literacy of the involved ageing users was very low.

Both the older adults and their caregivers were asked to participate in a training session dedicated to the Tailoring Environment and organized by each project partner handling a field trial. Each of such training sessions lasted a minimum of one hour, and it was structured into some steps. It started with a PowerPoint presentation about the tool. There was a video-based tutorial showing examples of specifying rules through the tool. Then, the elderly and their caregivers were asked to write some rules in natural language, where at least one rule consisted of two triggers and one action. They did the training under the supervision of the project's team member associated with their trial, who helped them only to build the first rule: afterwards, they were asked to proceed autonomously. At the end of the training, each subject received a user manual of the Tailoring Environment as reference information, to use during the field trial tests, just in case. Thus, during the training dedicated to the Tailoring Environment, some rules were already created. After the training, users had to use the tool in their everyday life, still relying on the possibility of asking for technical support in case of any inconvenience. During the field trial phase, new rules were added by the caregivers, as expected.

Out of the six people participating in the field trials, four caregivers were family members (i.e. sons or daughters). The remaining two were professional caregivers, as two older adults live alone in apartments for the elderly run by a foundation. Good collaboration with the project team occurred with caregivers, especially the informal ones. Regarding professional caregivers, in one case, the main caregiver changed job during the field trial. Below we describe the main aspects that characterised each trial, also providing information on the rules executed that has been possible to gather thanks to the analytic features of the platform.

In Trial 1, the subject was a female, 66 years old. The caregiver in charge of creating the rules is her son. In this trial, 74 rules were created in total, the highest number among all the trials. This was likely because the caregiver was a technology enthusiast, with some programming knowledge, and he felt very engaged by the project. The most used triggers were under the "Environment" dimension (about 66%), while the most used actions were under the "Appliances" category (about 89%). Most of the rules were set up to control the lights – mostly morning and evening lighting. Another rule involving the motion sensor was created so that the light could turn on if the elderly wakes up at night. In the second month, some rules were created involving generating alerts if the elderly's pulse rises above a specific value, or if a certain number of daily steps has not been reached. The elderly really enjoyed the lighting system, in particular the luminaire, which was used for better supporting her relaxation and reading. She also liked to play the cognitive stimulation games, which helped her to stay mentally active.

In Trial 2, the subject was a female, 85 years old, widow. She sometimes experiences episodes of memory loss, spatial disorientation, and falls. The caregiver is her daughter in law, who is a geriatrician. The elderly had never used modern devices (she had a classic phone and had never used a computer or a tablet before). In this trial, 9 rules were created in total. The reason for this low number of rules can be attributed to the fact that, in the initial parts of the trial, although the elderly looked confident that using system could have been interesting for her, she did not manage to comprehend her role in the project entirely. Afterwards, she progressively started to better accommodate with the system. The most used triggers were under the "Environment" dimension (about 85%); the most used actions were under the "Appliances" category (about 80%). In this trial, rules were created regarding the heart rate, smoke and gas detector for which an alarm was programmed to be sent to the caregiver. The use of the motion sensor and the hallway Light Strip was also part of several rules. Some rules were obtained by just editing previous ones, according to the needs and preferences of the user. The elderly in this trial appreciated the platform system overall, and the use of the serious game for stimulating her cognitive activity.

In Trial 3, the subject was a male, aged 81, quite autonomous in the management of his daily life. He participated with great enthusiasm and interest in the project. A nurse was the professional caregiver involved in this trial, in which 20 rules were created. The most used triggers were under the "Environment" dimension (about 74%); the most used actions were under the "Appliances" category (about 70%). At the beginning of the field trial phase, the subject and his main caregiver agreed to create rules for: i) automatic starting activating lights if there is no morning movement (to help him wake up in the morning); ii) automatic switching on the luminaire if there is low light at specific times (e.g. in the evening); iii) notifications on the tablet to remind the subject to do cognitive stimulation exercises, or to put eye drops, or to drink water at regular intervals. In the weeks following the installation of the system, some changes were made to the existing personalisation rules involving lights, to better support the elderly needs. The elderly used the smartwatch almost every day. After some initial difficulties with the tablet, he became increasingly familiar with it and found its use stimulating. He believed that the platform is useful for other seniors like him. He was so satisfied by it that he declared to be willing to participate in the next round of trials.

In Trial 4, the subject was a male, 83 years, widower. He had diabetes and walked with difficulty (with a cane), and took several medicines (e.g. for managing pressure and heart –related diseases). A professional caregiver was involved in this trial, where 27 rules were created. The most used triggers were under the "Environment" dimension (about 85%), while the most used actions were under the "Appliances" category (about 83%). At the beginning of the field trial, the caregiver was especially interested in sending a notification on the tablet to remind the elderly to measure his blood sugar. At the beginning, the patient, while accepting to be involved in the trial, declared not to really need the platform as he typically does not spend much time at home and when he comes back he usually watches TV and then goes straight to bed. The patient has a low education level, and also lacks familiarity with technology (indeed, the use of the tablet was also not simple for him at the beginning). In the first few weeks of the test, it was agreed that he would do the cognitive stimulation games together with the support of the caregiver. Thanks to this collaboration, in the end the elderly managed to do exercises on the tablet twice a week, and also to receive some notifications on the tablet. At the beginning, the caregiver was a bit sceptical about setting many new rules, since she was afraid that the changes would make the man more anxious. Despite this, during the trials, new rules were programmed, e.g. automatically switching on the bathroom light when the sensor detected movement near the bed during the night.

In Trial 5, the subject was a female, 78, widow. The caregiver in charge was her son, who lives on the other side of the city. In this trial, 23 rules were created. The most used triggers were under the "User" dimension (about 67%); the most used actions were under the "Alarms" category (about 60%). The rules that were built were mainly about remembering to take medicines and to do cognitive exercises on the tablet, and switching on/off the luminaire. Differently from other users, she wore the smartwatch quite regularly. During the field trials, her mood disorders sometimes had an impact on her sense of self-sufficiency (i.e. sometimes she stopped using the tablet or the application because she felt a bit demoralized when she encountered some difficulties). However, over time there was an improvement in her skills in using the system. She often exploited rules involving the coloured lights to solve some melancholy-related issues she suffered. Among the good effects obtained through the use of this platform, she mentioned that she really liked the possibility of setting up a (pink) colour for the lights. She declared that, when she experienced some episodes of sadness, such pink light had a very relaxing effect on her –without the need of assuming any medication, she highlighted. She also mentioned that the personalised reminders set up for prompting her to regularly use the game, not only had indubitable utility for improving her cognitive skills but also contributed in making her progressively learn to use tablets, which she realised as a key skill for opening further challenging opportunities on a more general level. At the end of the trials, she declared feeling happier, and willing to use the system in the future (especially the lights). She also declared that she would have suggested to other older adults to participate and try the platform. As a sign of her level of acceptance of the system, she agreed to participate in the next round of trials.

In Trial 6, the subject was a female, 80 years old, widow, who suffered from heart rate alteration. The caregiver was her son. The triggers most used were under the "User" dimension (nearly 100%), the actions most used were under the "Alarms" category (nearly 100%). The 6 rules created overall in this trial mainly involved the use of medicines. The reason for this low number of rules is associated with the fact that, unfortunately this elderly dropped out after less than a month. Beyond considering the smartwatch as too heavy for her, she declared that at the beginning she thought that the platform was more stimulating for her whereas, in the end, she felt it was not the case (e.g. she reported to have found the games too easy). She also declared to be totally autonomous in the management of her daily life. However, this reconstruction of events was not at all shared by her caregiver, who instead showed interest and willingness to proceed with the field trial for improving the quality of life of the cared person. In particular, it came out that the elderly's lack of awareness of her memory deficits made her believe that she did not need any help in her activities. In addition, some technical issues with installing the luminaire (which was a prototype from another research project) may also have contributed to make the subject a bit impatient and annoyed.

The table below summarises some aspects associated with each trial reported, thanks to the information gathered through the TAREME platform.

Trial N.	N. of rules	The trigger type most used	Most used action type	Rules
Trial 1	74	Environment: 66%	Appliances: 89%	Most rules were set up to control the lights. Another rule set up so that the light could turn on if the patient wakes up at night. Rules involving generating alerts if the user's pulse rises above a specific value, or if a certain number of daily steps has not been reached.
Trial 2	9	Environment: 85%	Appliances: 80%	Rules created regarded the heart rate, smoke and gas detector for which an alarm was programmed to be sent to the caregiver. The use of the motion sensor and the hallway light strip was also included in several rules.
Trial 3	20	Environment: 74%	Appliances: 70%	Rules have been created for: i)automatic starting the activation of lights scene if there is no morning movement (to help the cared person wake up in the morning); ii) automatic switching on the luminaire if

				there is low light at specific times in the evening; iii)notifications on the tablet to remind the subject to do cognitive stimulation exercises, or to put eye drops, or to drink water at regular intervals.
Trial 4	27	Environment: 85%	Appliances: 83%	Rules involving a notification on the tablet to remind the elderly to measure blood sugar. Another rule regarded automatically switching on the bathroom light at night when the sensor detected movement near the bed.
Trial 5	23	*User*: 67%	*Alarms*: 60%	Rules involving reminders set up for stimulating her to use regularly the game. Rules involving setting up coloured lights in particular moments of the day, to have relaxing effects.
Trial 6	6	*User*: 100%	*Alarms*: 100%	The rules created in this trial mainly involved the use of medicines.

Table 1: An Overview of personalization rules created and executed in the trials

Overall, 164 rules were created by users (as it has been shown in Figure 6). Such rules were triggered 15224 times in the various apartments. Most personalisation rules were programmed in the initial phase of the field trial. Nevertheless, during the trials, additional ones were added or resulted by better defining or adapting the rules initially programmed. The created rules referred mainly to lights and notification/alerts. In particular, they regarded: i) control the morning and evening lighting; ii) automatic switching on the lights to help the person wake up in the morning; iii) use the information coming from the motion sensor to activate the hallway LightStrips; iv) switch on the Luminaire in case of low light at predefined times; v) switch on the Luminaire at selected times. As for notification and alerts: i) generating alerts if the heart pulse rises above a certain threshold; ii) generating alerts if a certain number of steps is not reached; iii) generating alerts if smoke or gas is detected; iv) notification on the tablet to remind to take medicines or to measure blood sugar; v) notification on the tablet to remind to do cognitive games.

It is worth pointing out that often rule execution depends on user behaviour in indoor environments. While various research efforts have been put forward to address such issues, there are a few general solutions able to provide indoor localization stably and continuously, with limited costs. We found useful the use of the smartwatch since it can be worn with limited effort, so guaranteeing a continuous information flow. Besides, it can also be useful to distinguish between movements of different people inside the same home. However, the smartwatch model we used was large and a bit cumbersome, and needed to be recharged frequently because of the battery consumption caused by the location Bluetooth-based scanning. Thus, sometimes users forgot to use it and this decreased the set of events that could be detected in that period.

During the first contacts, the subjects chosen for the field trials and their caregivers (family members and formal caregivers), show, overall, interest in participating in the trial and experiencing the personalization possibilities offered by the platform through the devices and applications we made available to them. Although the project idea was generally highly welcomed at the beginning, the field trial phase then brought to light some aspects that influenced positively or negatively their attitude toward the system.

On the one hand, some people showed from the beginning and maintained until the end of the trial period a positive attitude and openness to exploiting the personalisation platform, and they overall seemed to be well receptive to the introduction of this new system in their houses and their routines. For instance, the lady involved in Trial 5 expressed very enthusiastic feelings from the initial phases up to the end of the trial. She decided to join the trials when she realised to suffer from some mild memory losses, which made her progressively isolating, with less willingness to go outside and also having a negative impact on her mood (i.e. more anxiety). After the end of the trials, the lady declared to feel more self-confident, more peaceful, with more willingness to do things and to go outside (she also declared that she would have recommended the use of this platform to others, as mentioned before).

On the other hand, some people expressed some scepticism and anxiety about the applications, appliances and devices we deployed in their home, and they seem not willing to accept some small modifications to their routines required to be meaningfully involved in the trials (e.g. to control the appliances through rules and not manually, to wear the smartwatch), or they struggled to understand completely the usefulness of the system in better supporting their activities. For instance, in Trial 6, while the caregiver was very interested in continuing using the platform, the elderly decided to leave the project after about one month. One of the contributing reasons was that at the beginning the Luminaire did not work well, therefore the needed external interventions made her nervous. However, also her lack of self-awareness of her cognitive deficits made her not perceive much need for help, since she believed that joining the project could have been more useful for people with a greater level of cognitive deficits.

In addition to maintaining the seniors' enthusiasm and personal openness towards the project, it was also very important to collaborate with their caregivers to keep high their interest in participating in the trials and to solve any issues that emerged. The caregivers generally thought highly about the potential of the system in improving the lives of the persons they care, as well as their own lives. For instance, they often mentioned the potential of becoming better aware of the situation of the elderly (e.g. at night and in general when away from elderly) with increased peace of mind for them, as well as the potential of relieving the burden associated with their tasks (e.g. by sending regular reminders for encouraging the elderly to play the cognitive games, by helping in better structuring elderly's life). Some of them expressed from the beginning enthusiasm toward the platform and the possibility to use it. However, there were also cases (for instance in Trial 4) where the (professional) caregiver expressed an initial fear of changing too much the routines of the elderly, mainly because that elderly has a low familiarity with technology. However, during the trials, we were able to see that the system was more and more adopted, following an 'incremental' approach: initially the platform was just used for sending notifications/reminders to elderly, then it started playing a more 'active' role (by controlling appliances in the house, i.e. the lights), which shows a more willing attitude of including the use of the platform in the elderly life (which also has the additional side effect of progressively increasing the elderly familiarity with technology). As an additional point on the side of caregivers, especially during the training, we noted that sometimes people completely unfamiliar with the trigger-action approach could not immediately understand its potentialities and quickly identify concrete situations in which to exploit it. However, when we provided them with concrete examples of relevant possibilities available to them, their interest increased, and they began thinking about possible extensions and refinements.

7 Discussion

In this section, we discuss aspects we deem useful for those interested in supporting personalization of IoT environments based on our work in designing, implementing and deploying a platform for this purpose, with a particular focus on the analytics features it provides about the actual use of the rules.

Monitoring rules for improving the specification of personalization. The presented rule analytics features not only can provide information about the personalization rules actually executed in users' contexts, which is useful for better understanding users' personalization preferences. They can also help in identifying personalized behaviour that might need improvement from the user's side, regarding e.g. the correctness of the automations expressed by the rules. For example, this could occur either when a rule is activated too many times, or when a rule has never been triggered, despite having been active for some time. For instance, in the trials, a user specified a rule involving the trigger "*amount of time a user is inside a room*" as the following condition: "*IF user time inside the bedroom is more than 1 hour*". Since that trigger was expressed as a condition, which can last for a long time, the concerned action was triggered many times, while it would have been sufficient to send the reminder once. So, in this case, the information about the (high) number of times a rule is triggered was helpful in detecting an error within the rule specification itself (namely: the use of a condition instead of an event, within the specification of a rule). Thus, the number of times a rule is triggered can be useful for identifying potentially 'wrong' rules, even though this number also depends on how long the rule has been active (it can be different if a rule has been triggered 1000 times in one day or in one month). For this reason, after the trials, we decided including additional information which shows how long a rule was active and the activation time intervals. However, it is also worth mentioning that, for some rules, not being executed within a specific (even long) time interval can be perfectly fine: for instance, this is the case of a rule that is expected to send an alarm after detecting a risky situation (e.g. when a gas leakage occurs).

'Incremental' rule definition. By ordering the rules created in each trial, we noticed that clusters of quite similar rules were created in a limited interval of time by users. This makes us think that people use a 'conservative', step-wise refinement approach to build rules, and they preferred saving distinct (even similar rules), instead of deleting previously created ones. One side-related effect of this behaviour is that rule repository could soon become unnecessarily large, as they include groups of rules that are very similar to each other. For the future, we can plan to have better support for managing such rule groups, e.g. by highlighting to users the rules that have some similarities. In other cases, analysing the executed rules was also useful to discover that sometimes caregivers initially tended to limit the use of the platform for just sending reminders and notifications, while in a next phase they started exploiting rules that actually controlled some appliances in the house, thus showing signs of a better attitude towards the adoption of the platform, over time.

Make a set of pre-defined rules available. In the trials a good number of rules was created. However, people unfamiliar with the trigger-action approach could not immediately understand its potentialities and identify concrete situations in which to exploit it. However, when we provided them with examples of the various possibilities that could be relevant for them, their interest increased, and they began thinking about possible extensions. To prevent burdening users with manually specifying rules, RuleSelector [34] provides a solution in which shortlists of relevant rules based on the user behaviour patterns are proposed. However, while such solution can work well with automations for personal smartphones, in more open and flexible environments such as IoT applications, it can be problematic to generate rules by only analysing the past user's behaviour. Thus, we think more appropriate to identify sets of potential 'pre-defined' rules together with domain experts and end-users, aimed to encourage re-use of rules. For this purpose, our environment provides a public space from where each user can import those rules which are deemed most suitable.

Installing the platform – some problems are discovered only during actual deployment. Even though the deployment of the system in the different trials was in the end overall satisfactory, installing the platform in the various trials was not free from hurdles and unexpected complications. The configuration of the various devices and sensors had to consider the specific configuration and also the layout of the apartments in which the system had to be deployed, since in each flat there was a specific and defined physical space for placing the equipment (and for setting up the related electricity/network connections). Considering that such flats are the home of real people, the installation was planned to shorten times and have the minimal impact in terms of changes in the homes, in order not to cause disturbances to their residents. We also planned a structured pre-test phase in which we prepared some training material to be used by caregivers/elderly, and an installation guide to be used by people who were expected to help in the installation, in order to enable them to have reference material for solving possible issues. The installation was done by members of the project who were the contact points for the patients involved in the trials. As such, they often were members of end-user organisations involved in the project, and therefore they often did not have specific technical competences. We also provided continuous support for possible installation issues occurring during the trials. Despite all such efforts, the installation of the platform with the associated objects and sensors in the field trial apartments encountered some difficulties, which were discovered only during the real deployment. The appliance that raised most issues was the luminaire, the only non-commercial device that was included in the trials, which is still a prototype. For instance, one issue was connected with the fact that the luminaire, in spite of working well during the tests done in the labs, when deployed in the actual elderly's homes, in some cases it became no longer reachable because it used a proprietary, unstable communication protocol.

Events, conditions and their compositions. In the proposed environment, we overcome one limitation of tools (such as IFTTT) that allow compositions of only one trigger and one action. We provide the possibility to compose triggers that gather information from different sensors and services, and to indicate actions that have effects on different objects and applications. Besides, the platform clearly supports the distinction between events and conditions. However, we noted that in some cases people specified rules that still had wrong combinations of triggers, e.g. a AND composition of multiple events, which results in a rather unlikely event. Thus, we have started to introduce further control to prevent the creation of meaningless trigger compositions.

Supporting the occurrence of actions when something has not happened yet. With the proposed system, we support a situation that can be very useful when people want to activate an action when an event has *not* occurred within a specific period. This possibility can be particularly interesting in various domains. For instance, it can be relevant for applications providing remote monitoring and assistance to the elderly because they can present more frequent episodes of memory lacks and therefore the possibility of not doing actions can be higher. However, it can also be useful in industrial settings to verify that specific actions have been carried in the production process, and therefore be able to send suitable actions (e.g. alarms) when it is not the case.

8 Conclusions and Future Work

In this paper, we have presented a EUD platform that provides support for various phases in the customisation process. In particular, in this article we focus on the support it provides for executing and analysing personalisation rules in IoT scenarios. The platform includes a Tailoring Environment through which end users can specify their trigger-action personalisation rules, and it also offers the middleware necessary to have such rules actually working in real scenarios. Analytics features have also been developed and integrated to allow relevant stakeholders (e.g. domain experts) to get more insights about the performance of the automations designed by end users, and their actual personalisation needs. We also report on the deployment of the platform in the AAL domain, in six trials run in two European countries. From the gathered experience, we draw several indications we deem useful for those interested in supporting personalization of IoT environments regarding designing, implementing and deploying a platform for this purpose.

While in this paper we report on trials of the platform carried out in the AAL domain, it is worth pointing out that it can be used in several distinct domains (such as smart retail, industry 4.0, museums). Indeed, on the one hand, there is no specific limit to the number of applications or sensors that can be connected to the platform. On the other hand, for the most part, customization of this platform for a specific application sector requires only some modifications in a limited subset of platform modules, as it would mainly mean developing relevant Context Delegates (to receive the information from the available, associated sensors and properly updating the Context Server), and adding suitable interpreters of the actions to execute within the involved applications. After we successfully managed to install the platform in the house of the involved older adults, we realized that the proposed platform can effectively enable many personalization scenarios. The trials were also useful to provide indications on the parts that need further refinements to better provide context-dependent support (such as the accuracy of the indoor localization), which are considered for preparing future, more extended deployment and adoption of the proposed technological platform.

As future work, we also plan to introduce intelligent support aiming to simplify the identification and specification of personalised behaviour by providing end-users with dynamic recommendations of rules coordinating the joint behaviour of devices, sensors and actuators available in their environments equipped with IoT technology.

ACKNOWLEDGMENTS
We gratefully thank support from the <anonymous> Project X and <anonymous> Project Y.

REFERENCES
[1] Pierre A. Akiki, Arosha K. Bandara, and Yijun Yu. 2017. Visual Simple Transformations: Empowering End-Users to Wire Internet of Things Objects. ACM Trans. Comput.-Hum. Interact. 24, 2, Article 10 (April 2017), 43 pages. DOI: https://doi.org/10.1145/3057857

[2] Carmelo Ardito, Giuseppe Desolda, Rosa Lanzilotti, Alessio Malizia, Maristella Matera. 2019. Analysing trade-offs in frameworks for the design of smart environments. Behaviour & Information Technology 39 (1), 47-71

[3] Carmelo Ardito, Paolo Buono, Giuseppe Desolda, Maristella Matera, (2018). From smart objects to smart experiences: An end-user development approach. International Journal Human-Computer Studies, vol. 113, p. 1-18, ISSN: 1071-5819, doi: 10.1016/j.ijhcs.2017.12.002

[4] Barbara Rita Barricelli, Fabio Cassano, Daniela Fogli, Antonio Piccinno: End-user development, end-user programming and end-user software engineering: A systematic mapping study. Journal of Systems and Software 149: 101-137 (2019)

[5] Barbara Rita Barricelli, Stefano Valtolina, A Visual Language and Interactive System for End-User Development of Internet of Things Ecosystems. J. Vis. Lang. Comput., 40 (2017), pp. 1-19

[6] Iulia Bastys, Musard Balliu, and Andrei Sabelfeld. 2018. If This Then What?: Controlling Flows in IoT Apps. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18). ACM, New York, NY, USA, 1102-1119. DOI: https://doi.org/10.1145/3243734.3243841

[7] Margaret Burnett, Brad Myers, Future of end-user software engineering: beyond the silos, Proceedings of the on Future of Software Engineering, 2014, 201-21, ACM.

[8] Will Brackenbury, Abhimanyu Deora, Jillian Ritchey, Jason Vallee, Weijia He, Guan Wang, Michael L. Littman, Blase Ur. 2019. How Users Interpret Bugs in Trigger-Action Programming. In CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland Uk. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300782

[9] Danilo Caivano, Daniela Fogli, Rosa Lanzilotti, Antonio Piccinno, Fabio Cassano, Supporting End Users to Control Their Smart Home: Design Implications from a Literature Review and an Empirical Investigation, Journal of Systems and Software 144, 295-313, 2018

[10] Nico Castelli, Corinna Ogonowski, Timo Jakobi, Martin Stein, Gunnar Stevens, and Volker Wulf. 2017. What Happened in my Home?: An End-User Development Approach for Smart Home Data Visualization. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). ACM, New York, NY, USA, 853-866. DOI: https://doi.org/10.1145/3025453.3025485

[11] Luca Corcella, Marco Manca, Fabio Paternò, Carmen Santoro: A visual tool for analysing IoT trigger/action programming. In: Bogdan, C., Kuusinen, K., Lárusdóttir, M.K., Palanque, P., Winckler, M. (eds.) HCSE 2018. LNCS, vol. 11262, pp. 189–206. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05909-5_11

[12] Fulvio Corno, Luigi De Russis, and Alberto Monge Roffarello. 2019. Empowering End Users in Debugging Trigger-Action Rules. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 388, 1–13. DOI:https://doi.org/10.1145/3290605.3300618

[13] Joelle Coutaz, James L. Crowley. A first person experience with end-user development for smart home. IEEE Pervasive Comput., 15 (2) (2016). May-June 2016: 26:39

[14] Luigi De Russis, Fulvio Corno. HomeRules: a tangible end-user programming interface for smart homes. CHI Ext. Abstr. (2015), pp. 2109-2114

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

[16] Chris Elsden, Tom Feltwell, Shaun W. Lawson, John Vines: Recipes for Programmable Money. CHI 2019: 251

[17] Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. 2017. Personalization of Context-Dependent Applications Through Trigger-Action Rules. ACM Trans. Comput.-Hum. Interact. 24, 2, Article 14 (April 2017), 33 pages. DOI: https://doi.org/10.1145/3057861

[18] Christos Goumopoulos and Irene Mavrommati, A framework for pervasive computing applications based on smart objects and end user development, The Journal of Systems and Software, Volume 162, April 2020, 110496. https://doi.org/10.1016/j.jss.2019.110496

[19] Valentina Grigoreanu, Margaret Burnett, Susan Wiedenbeck, Jill Cao, Kyle Rector, and Irwin Kwan. 2012. End-user Debugging Strategies: A Sensemaking Perspective. ACM Transaction on Computer-Human Interaction 19, 1, Article 5 (May 2012), 28 pages. https://doi.org/10.1145/2147783.2147788

[20] Sumit Gulwani and Mark Marron. NLyze: Interactive programming by natural language for spreadsheet data analysis and manipulation. In Proc. of: International Conference on Management of Data, pp. 803–814. Association for Computing Machinery (2014).

[21] Weijia He, Jesse Martinez, Roshni Padhi, Lefan Zhang, Blase Ur. When Smart Devices Are Stupid: Negative Experiences Using Home Smart Devices. 2019 IEEE Security and Privacy Workshops (SPW)

[22] 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, New York, NY, USA, 215-225. DOI: https://doi.org/10.1145/2750858.2805830

[23] Timo Jakobi, Gunnar Stevens, Nico Castelli, Corinna Ogonowski, Florian Schaub, Nils Vindice, Dave Randall, Peter Tolmie, Volker Wulf. Evolving Needs in IoT Control and Accountability: A Longitudinal Study on Smart Home Intelligibility. Vol. 2, No.4, Article 171. Journal Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2018

[24] Nicola Leonardi, Marco Manca, Fabio Paternò, and Carmen Santoro. 2019. Trigger-Action Programming for Personalising Humanoid Robot Behaviour. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). ACM, New York, NY, USA, Paper 445, 13 pages. DOI: https://doi.org/10.1145/3290605.3300675

[25] Toby Jia-Jun Li, Yuanchun Li, Fanglin Chen, Brad A. Myers. Programming IoT devices by demonstration using mobile apps. In S. Barbosa, P. Markopoulos, F. Paternò, S. Stumpf, S. Valtolina (Eds.),

End-User Development. IS-EUD 2017. Lecture Notes in Computer Science, Vol 10303, Springer, Cham (2017).

[26] Marco Manca, Fabio Paternò, Carmen Santoro, Luca Corcella: Supporting end-user debugging of trigger-action rules for IoT applications. Int. J. Hum. Comput. Stud. 123: 56-69 (2019)

[27] Sarah Mennicken, David Kim, and Elaine May Huang. 2016. Integrating the Smart Home into the Digital Calendar. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 5958-5969. DOI: https://doi.org/10.1145/2858036.2858168

[28] Georgios Metaxas, Panos Markopoulos. Natural contextual reasoning for end users ACM Trans. Comput.-Human Interact. (ACM TOCHI), Vol.24 (2) (2017)

[29] Xianghang Mi, Feng Qian, Ying Zhang, and XiaoFeng Wang. 2017. An empirical characterization of IFTTT: ecosystem, usage, and performance. In Proceedings of the 2017 Internet Measurement Conference (IMC '17). ACM, New York, NY, USA, 398-404. DOI: https://doi.org/10.1145/3131365.3131369

[30] Mateusz Mikusz, Sarah Clinch, Rachel Jones, Mike Harding, Christopher Winstanley, Nigel Davies. Repurposing web analytics to support the IoT. IEEE Comput. 48, 42–49 (2015)

[31] Brad A. Myers, Andrew J. Ko, Chris Scaffidi, Stephen Oney, YoungSeok Yoon, Kerry Chang, Mary Beth Kery and Toby Jia-Jun Li. Making End User Development More Natural. F. Paternò, V. Wulf (Eds.), New Perspectives in End-User Development, Springer, Cham (2017)

[32] Michael Rietzler, Julia Greim, Marcel Walch, Florian Schaub, Bjorn Wiedersheim, and Michael Weber. 2013. homeBLOX: Introducing process-driven home automation. In Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct (UbiComp'13). ACM, 801–808. DOI:http://dx.doi.org/10.1145/2494091.2497321

[33] Eugene Siow, Thanassis Tiropanis, and Wendy Hall. 2018. Analytics for the Internet of Things: A Survey. ACM Comput. Surv. 51, 4, Article 74 (July 2018), 36 pages. DOI: https://doi.org/10.1145/3204947

[34] Vijay Srinivasan, Christian Koehler, and Hongxia Jin. 2018. RuleSelector: Selecting Conditional Action Rules from User Behavior Patterns. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2, 1, Article 35 (March 2018), 34 pages. DOI: https://doi.org/10.1145/3191767

[35] Milijana Surbatovich, Jassim Aljuraidan, Lujo Bauer, Anupam Das, and Limin Jia. 2017. Some Recipes Can Do More Than Spoil Your Appetite: Analyzing the Security and Privacy Risks of IFTTT Recipes. In Proceedings of the 26th International Conference on World Wide Web (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1501-1510. DOI: https://doi.org/10.1145/3038912.3052709

[36] Daniel Tetteroo and Panos Markopoulos (2015). A review of research methods in end user development. In Díaz P., Pipek V., Ardito C., Jensen C., Aedo I., Boden A. (Eds.), End-user dev (pp. 58–75). Springer International Publishing.

[37] Daniel Tetteroo and Panos Markopoulos: EUD Survival "in the Wild": Evaluation Challenges for Field Deployments and How to Address Them. New Perspectives in End-User Development 2017: 207-229

[38] Daniel Tetteroo, Paul Vreugdenhil, Ivor Grisel, Marc Michielsen, Els Kuppens, Diana Vanmulken, and Panos Markopoulos. 2015. Lessons Learnt from Deploying an End-User Development Platform for Physical Rehabilitation. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 4133-4142. DOI: https://doi.org/10.1145/2702123.2702504

[39] Jordan Tewell, Dympna O'sullivan, Neil Maiden, James Lockerbie, and Simone Stumpf. 2019. Monitoring meaningful activities using small low-cost devices in a smart home. Personal Ubiquitous Comput. 23, 2 (April 2019), 339-357. DOI: https://doi.org/10.1007/s00779-019-01223-2

[40] Tommaso Turchi, Alessio Malizia, Alan Dix: TAPAS: A tangible End-User Development tool supporting the repurposing of Pervasive Displays. J. Vis. Lang. Comput. 39: 66-77 (2017)

[41] Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. 2014. 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. https://doi.org/10.1145/2556288.2557420

[42] Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, Jiyun Lee, Sarah Mennicken, Noah Picard, Diane Schulze, and Michael Littman. 2016. Trigger-Action Programming in the Wild: An Analysis of 200,000 IFTTT Recipes. In Proceedings of the 34rd Annual ACM Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 3227–3231. https://doi.org/10.1145/2858036.2858556

[43] Qi Wang, Pubali Datta, Wei Yang, Si Liu, Carl A. Gunter, Adam Bates. Charting the Attack Surface of Trigger-Action IoT Platforms ACM Conference on Computer and Communications Security. July, 2019

[44] David Weintrop, Afsoon Afzal, Jean Salac, Patrick Francis, Boyang Li, David C. Shepherd, and Diana Franklin. Evaluating CoBlox: a comparative study of robotics programming environments for adult novices. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18), New York, NY, USA, ACM (2018), p. 12. Paper 366

[45] Svetlana Yarosh, Pamela Zave. Locked or Not?: Mental models of IoT feature interaction Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17), New York, NY, USA, ACM (2017), pp. 2993-2997 http://dx.doi.org/10.1145/3025453.3025617