# Towards Intelligent Personalization of IoT Platforms

**Fabio Paternò, Sadi Alawadi**
HIIS Laboratory, CNR-ISTI
Pisa, Italy
{fabio.paterno, sadi.alawadi}@isti.cnr.it

## ABSTRACT

Trigger-action programming is an emerging paradigm for personalizing the behaviour of Internet of Things applications and services. In this area some recent research efforts have been dedicated to designing environments able to support people without programming experience in specifying the desired personalization rules. Little attention has been paid to how to provide intelligent support in identifying relevant rules. In this position paper we discuss the possible approaches to making such personalization platforms able to exploit the data collected on the behaviour of the users as well as the surrounding devices and things.

## CCS CONCEPTS

**•Human-centered computing~Human computer interaction (HCI)** • Computer systems organization~External interfaces for robotics • Software and its engineering~Software notations and tools

## KEYWORDS

End User Development; Internet of Things; Trigger-Action Programming

**ACM Reference format:**
Fabio Paternò and Sadi Alawadi 2019. Towards Intelligent Personalization of IoT Platforms. *In Joint Proceedings of the ACM IUI 2019 Workshops, Los Angeles, USA, March 20, 2019,* 3 pages.

## 1 INTRODUCTION

The Internet of Things (IoT) is the network of objects in our daily life (such as lights, refrigerators, car components, medical devices, dog collars, etc.) that can send or receive information with other devices on the network. These objects include sensors and actuators of various kinds and can interact with each other, with human beings and with the environment to exchange data in order to react to real-world events, trigger actions and activate services. They are increasingly used in many sectors: home, retail, industry, agriculture, and so forth. Consequently, we use our applications more and more in dynamic contexts in terms of services, devices, objects and people where many events can occur.

Users' activities can vary quickly and some of them cannot even be anticipated when designing interactive applications, but are only discovered during their actual use. It may be very difficult for developers to predict all possible context-dependent scenarios, because there may be unforeseen (at design time) requirements that need to be supported when the application is actually used. Indeed, we are all different from each other in terms of culture, abilities, interests, goals. The increasing pervasiveness of such technologies implies that they are more sensitive than ever to such differences. Thus, end-user development (EUD) approaches [2], which aim to allow people without programming experience to create or modify their applications, have become particularly relevant to address emerging IoT scenarios.

Trigger-action programming [1] is a development paradigm to support management of the dynamic situations characterizing the advent of the IoT. It is based on the specification of rules whose structure is simple: it indicates that if/when something happens in terms of events and/or conditions, then the desired effect should occur. An example of an environment supporting this approach is IFTTT. It is a Web and mobile environment that allows users to create rules, called «applets», in the form if trigger then action, where triggers and actions can be chosen from existing services (for example, Facebook, Evernote, Weather, Dropbox, etc.). It is estimated that 320000 applets involving 400 service providers have been installed more than 20 million times [5]. Services involve peripherals, hubs, wearable devices, social networks, messaging, .... However, IFTTT has some limitations. It supports applets composed of only one trigger and one action, while previous work [6] has reported a test with 226-participants that has shown that even people without programming experience can easily use rules with multiple triggers and actions by extending the IFTTT language. It does not distinguish between events and conditions. It is not easy to extend the list of the connected applications,

IUI Workshops'19, March 20, 2019, Los Angeles, USA.

and it shows long lists of potential channels to compose, where it is easy to get lost. Some recent research efforts aim to overcome such limitations [1, 4].

## 2 EXAMPLE PERSONALIZATION PLATFORM

An example personalization platform is reported in [3]: it is composed of a set of tools that allows people without programming experience to dynamically personalize their interactive applications, smart things, and humanoid robots in order to best suit their needs in specific contexts of use. It relies on a middleware (context manager) that is composed of a server that can collect, model and interpret the data generated by the various sensors and objects, which are collected through context delegates, small software that convert raw data into a more structured format that can be better managed by the context server. The platform includes a personalization rule editor that enables even people without programming experience to easily define their specific rules in order to adapt the behaviour of the available objects, devices, and applications to their preferences. For this purpose, it shows the possible triggers that are logically structured in terms of the aspects they refer to (user, environment, technology), and the possible consequent actions (reminders, alarms, user interface modifications, changes in the state of some appliances, etc.). The platform also includes a rule manager, which receives the personalization rules to execute, and subscribes to the context manager to be informed when some events or conditions associated to any of such rules are verified, so that the consequent actions can be executed. It has been extended in order to also support the personalization of humanoid robots, which can be considered an integrated set of sensors and actuators, with the additional possibility to perform human-like behaviour.

The platform can be applied in several domains exploiting IoT technologies (e.g. smart retail, industry 4.0, home, health, etc.). For example, we are organising trials of this type of personalization platform for the PETAL project in the Ambient Assisted Living domain. In this context, the target users are elderly with mild cognitive impairments and their formal and informal caregivers. The older adults have a tablet for their interactions with applications supporting cognitive stimulation exercises and providing information useful for their daily life. They also wear a smartwatch able to connect with Bluetooth and Wifi at the same time. This device is exploited to detect personal data (e.g. step counters and heart rate), and user's indoor position (with the support of Bluetooth proximity beacons), these sensed data are sent to the context manager via Wifi connection. Other sensors include beacons with accelerometers to detect use of doors, windows, cupboards, objects, and sensors to detect motion, light, indoor and outdoor temperature, humidity. In addition, a number of lights that can be controlled in terms of colour, intensity, temperature, duration will be used as well for providing stimuli in order to help the older adults to better organise their daily activities. Thus, the lights can set scenes with activation and relaxation effects depending on the period of the day and the state of the user.

## 3 INTRODUCING INTELLIGENCE IN THE PERSONALIZATION PLATFORM

The massive amount of the data generated by this type of personalization platform can be processed in a way to add intelligence to the platform. Machine Learning (ML) algorithms can be used to achieve this goal. In this case, ML algorithms should be exploited to build knowledge by processing the available data with the goal to allow the platform to take dynamic real time initiatives based on what has been learnt from the historical data.

In order to introduce in this type of personalization platforms the ability to exploit the data related to the behaviour of users, devices and objects with the goal to automatically identify relevant personalization rules, there are two possible types of relevant data (see Figure 1):

- Data concerning the user behaviour (such as movements, interactions with objects and applications, ...), and contextual conditions (such as time, weather, light, ...);
- Data concerning the personalization rules that have been created and executed beforehand.

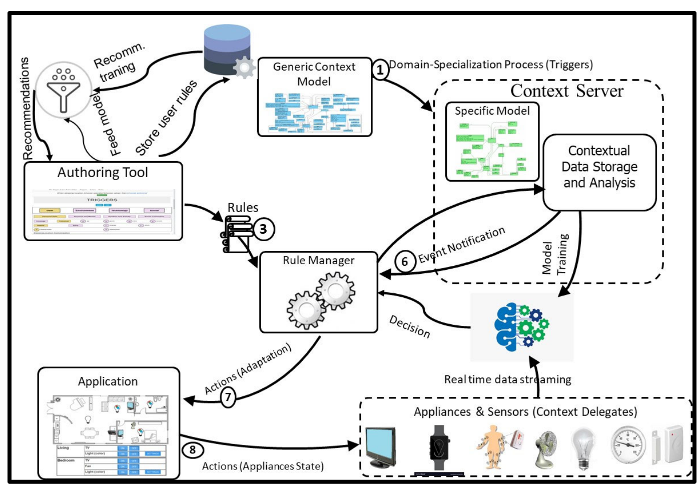


**Figure 1. Personalization platform with indication of the areas where intelligent support can be introduced.**

**Behavioural and contextual data**

In this type of platform, a machine learning approach should be able to identify in which contextual situations it is preferred that some actions occur. Based on the historical data collected by the context manager it is possible to train a model, which is then used to identify some rules. This type of approach can be effective when the possible events and actions are limited. For example, an intelligent thermostat is able to detect the room temperature at which the heating system should be turned on based on the user's previous choices. Thus, in this case the data are when the user turned on the thermostat and the temperature at that time. The consequent action in this case is to turn the thermostat on. A more general solution is rather difficult because it implies the ability to monitor in a continuous and reliable way many possible relevant contextual aspects and actions performed (corresponding to changes of the appliances status or commands sent to applications). Thus, for example, if the goal is to personalize the use of the lights, then we should have tools able to monitor various aspects (time, user position, weather, …) and actual lights uses in order to identify possible routines and preferences in associating them.

One further issue is how to exploit the rules identified through machine learning. Indeed, previous studies [7] have identified various issues in intelligent systems for the home, such as the learning system failing to understand user intent or the system's behaviour being hard to understand.

**Rules-related data**

In this case it can be possible to design a personalization rule recommendation system based on rules previously provided according to two possible approaches: collaborative or content-based filtering. In collaborative filtering the basic idea is that if user A has specified a rule, which is also indicated by user B, then there is a good probability that other rules specified by user B can still be relevant for user A. Thus, it is sufficient to monitor the rules indicated by various users and then apply this approach to identify what rules to suggest. In the case of content-based filtering it is possible to consider various types of recommendations. Some can be obtained through *generalization* of the content of some part of the existing rules. Thus, for example if there is a rule that says that when the user enters the bedroom then the lights should be on, one possible suggestion obtained through generalization can be that when the user enters any room, then the lights should be on. Another type of recommendation can be obtained by trigger *refinement*, which means narrowing the situations when the trigger should be fired by adding conditions that make the rules more suitable to meet more precisely specified needs.

Other recommendations can be based on the actual users' behaviour and their *preferences*. This means exploiting a specific case of context-aware recommendation systems [8]. For example, they can be *device-oriented*: if the user prefers to use some specific device, then it can be meaningful to suggest rules that exploit it, for example for sending alarms. One further aspect is to have *location-dependent recommendations*, which aim to provide suggestions when the users are in a specific area that they seem to prefer. Likewise, they can also be *time-oriented* recommendations, which are rules with triggers associated with specific periods of the day when the user is more inclined to receive information (e.g. reminders).

## 4 CONCLUSIONS

Trigger-action programming captures various aspects that characterize the emerging need to support personalization of IoT environments. EUD approaches can empower even people without programming experiences to customize such applications. Although intelligent support can be useful in the perspective to augment these EUD approaches based on the analysis of the data concerning the actual user behaviour, it needs to be carefully designed in order to obtain meaningful results.

In this position paper we provide some indications about what possible approaches to data collection and analysis can help us to obtain mixed-initiative, usable platforms for personalization of daily IoT environments in such a way as to enhance the user experience.

## REFERENCES

[1] G Ghiani, M Manca, F Paternò and C Santoro (2017). Personalization of Context-dependent Applications through Trigger-Action Rules. ACM TOCHI, Vol.24, Issue 2, Article N.14, April 2017.

[2] H Lieberman, F Paternò, M Klann and V Wulf (2017). "End-user development: An emerging paradigm" In: "End-user development", pp. 1-8, Springer, 2006.

[3] N.Leonardi, M.Manca, F.Paternò, C.Santoro, Trigger-Action Programming for Personalising Humanoid Robot Behaviour, Glasgow, 2019 ACM CHI'19.

[4] P Markopoulos, J Nichols, F Paternò and V Pipek (2017). End-User Development for the Internet of Things, ACM Transactions on Computer-Human Interaction (TOCHI) Volume 24 Issue 2, 9, 2017

[5] Mi, X., Qian, F., Zhang, Y., Wang, X.F., 2017. An Empirical Characterization of IFTTT: Ecosystem, Usage, and Performance, Proceedings of Internet Measurement Conference (IMC) '17,

[6] B Ur, E McManus, MPY Ho and ML Littman (2014). Practical trigger-action programming in the smart home. CHI 2014: 803-812

[7] R. Yang, M. W Newman Learning from a learning thermostat: lessons for intelligent systems for the home, 2013 ACM international joint conference on Pervasive and ubiquitous computing, pp. 93-102

[8] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, 2011, Recommender systems handbook, pp. 217-253, Spring