Analyzing Trigger-Action Programming for Personalization of Robot Behaviour in IoT Environments

Marco Manca, Fabio Paternò, Carmen Santoro[1]

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

Abstract. The rising spread of humanoid robots in various settings of human life, and their increasing affordability, as well as the massive adoption of the Internet of Things (IoT) in various scenarios have made End User Development (EUD) for robotic and IoT applications an interesting research direction. In particular, in the EUD field, trigger-action rules have become popular for their simple structure, which enables users to create rules to implement their desired personalization. Such rules can be a precious source of information for various goals: understanding the aspects people are most interested in, the types of routines they would like to have, the kind of support/automation they would expect from the robot, and the environment in which the robot is immersed. However, since the number of rules that could be generated using such EUD tools could be significant, manual analysis of rules does not seem a viable solution. In this paper we discuss how a visual analytics tool supporting filtering, exploration and analysis of data generated by a EUD tool for programming humanoid robots immersed in IoT environments can be helpful for deriving relevant information associated with the personalization that users express through rules. The analysis can provide designers and developers of EUD tools and associated customizable applications with useful insights for improving the tools and the robotic applications themselves, and facilitate their adoption.

Keywords: EUD Analytics, Personalization, IoT, Robots.

1 Introduction

The massive adoption of the Internet of Things (IoT) in various scenarios ranging from smart home devices and smart cities to medical and healthcare applications, as well as the users' increasing need to personalise the application they use, have made end-user personalization of IoT environments the focus of several research contributions. The TOCHI special issue on EUD for IoT [11] provides a good introduction to this field. Some EUD tools are also becoming popular among users (see e.g. the IFTTT tool[1]), who are likely to rely on such rules on a daily basis. Recently, also general-purpose

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

humanoid robots are becoming increasingly widespread in numerous settings since they can help users in many real-world tasks and their cost is becoming increasingly affordable. As such, a novel trend is emerging in the EUD area, focused on EUD solutions (e.g. with the support of personalization rule editors) that address customization of robot applications by users who are not professional developers in IoT settings.

Due to the pervasiveness and versatility of use of IoT/robot applications in numerous contexts, the interactions that *end users* carry out with EUD tools can represent a precious source of information e.g. for *developers* of EUD tools, to understand the personalization aspects which people are focusing on most, the types of routines they would like to put in place, the types of automations they would expect from robots and surrounding environments. However, since the amount of information that could be generated by such EUD tools could be significant [12, 20], manual analysis of the data produced as a consequence of end user interaction with EUD tools does not seem a viable approach to interpret such data and quickly make sense of user activities, interests, and preferred routines. To this regard, interactive visual analytics solutions can be useful to obtain valuable insights into large data sets generated by applications used in IoT settings. By providing EUD tool developers with visualization and analysis of relationships in the dataset (e.g. through interactive dashboards), new insights can be gained in an interactive manner, and they consequently should be able to make better decisions accordingly. Indeed, the provided visualizations are expected to make the dataset clearer by presenting information in intuitive and user-friendly ways, opening up the possibility to delve into data to unveil novel insights that can be translated into opportunities for improvement.

In this paper we present a contribution discussing how a visual analytics tool can be useful to support analysis, exploration and filtering of relevant data derived from the analysis of the interactions done by end users with a EUD tool aimed at personalizing the behaviour of a robot immersed in a IoT environment. While in this paper we consider a consumer (home) scenario for the evaluation, the presented approach could also be applied in other contexts of use (e.g. work settings). We propose a method aiming to derive higher-level information such as usage patterns followed by users while interacting with the tool, the types of rules that users were most interested in creating by using the tool, the most popular trigger and action types used, so as to facilitate the supported interactions and also the adoption of the EUD tool and the associated applications. The method also includes an interactive module that prompts users with relevant questions to gather feedback at the end of the tool use. In the paper we also discuss the feedback we gathered from a group of end users without knowledge in IoT programming who were asked to use an EUD trigger-action tool for creating various rules aimed at personalizing the behaviour of a robot immersed in a IoT environment.

More in detail, after introducing the context and the motivations of this work, in the next section we describe significant related work in relevant areas, then we describe the method we developed and the associated tool support for analysis of rules created by using a trigger-action rule editor. Afterwards, we report on a study we carried out to assess the potentialities of the presented method, discuss the results, then conclude with final remarks and viable directions for future work.

2 Related Work

2.1 Trigger-Action Programming

In recent years, interest in using trigger-action programming for IoT environments has considerably increased. Various studies have been carried out to investigate how to propose this type of solution and its benefits [3, 5, 6, 11], and several issues have been discussed. For example, whether the single trigger/single action solution proposed by IFTTT is the most effective one [19].

In parallel, another emerging trend is the rapid diffusion of robots, and the associated need to support even people without programming experience to modify their behaviour. Various EUD approaches have been pursued to make easy the development of robotics applications.

One technique is programming by demonstration [1], in which users do not have to explicitly program each detail of the robot behaviour, but they have just to demonstrate how to achieve the robotic task by providing examples through which the robot should learn the new expected behaviour. Another solution is the use of visual toolkits with more user-friendly interfaces that guarantee rapid interaction development without aiming to optimize the final resulting performance. In this area we can distinguish approaches that use iconic data flow representations of robot functionalities (such as Choregraphe [15]), and those that use block-based languages introduced by Scratch [16], which have also been applied in other domains. An example of a visual programming tool for user-friendly, rapid programming manipulators robots is in [10]. The system is designed to let non-roboticists and roboticists alike program end-to-end manipulation tasks. The authors present findings from an observational user study in which ten non-roboticist programmers were able to learn the system and program a robot to do manipulation tasks. Buchina et al. [2] propose a design of a Web-based programming interface that aims to support end-users with different backgrounds to program robots using natural language.

Still in the attempt of lowering the barriers to programming robots, and then make it more accessible and intuitive for novice programmers, Weintrop et al. [21] present a block-based interface (CoBlox) for programming a one-armed industrial robot. The results show that participants using the block-based interface successfully implemented programs faster with no loss in accuracy while reporting higher scores for usability, learnability, and overall satisfaction. However, they considered a rather limited scenario, with one-armed industrial robots, and the participants were asked to program a "pick and place" routine. In [8] an end user development solution based on trigger-action personalization rules able to support programming of robots immersed in IoT environments has been presented, showing the potential for using trigger-action programming to make Pepper robot[2] behaviour personalization possible even for people who are not professional software developers.

[2] https://www.softbankrobotics.com/emea/en/pepper

Thus, the trigger-action programming paradigm seems a promising approach to address in an integrated way the need of personalization. We can foresee its adoption and use by many users in different contexts. Indeed, for example 320,000 IFTTT rules involving 400 service providers have already been installed more than 20 million times [12]. This trend poses new issues and requires novel approaches to support developers of trigger-action based tools and IoT and robotic applications to analyse the actual use of their tools.

2.2 Visual Analytics of Rule-Based Behaviour

Visual analytics tools have been often used for analysing logs associated with user interactions with Web applications. Indeed, interaction logs provide detailed information about the sequence of steps which the users take in order to reach their goals and then provide useful information for revealing various pieces of relevant information, such as patterns followed by users while interacting with Web applications. As such, one typical use of the information contained in logs of user interactions is for usability studies. For instance, Harms and Grabowsky [9] proposed transforming the recorded user interaction in task trees that are then checked to identify usability issues. The goal of such contributions is to identify a method to record user interactions and then further analyse the logs in order to highlight usability problems. HistoryViewer [17] is a system that aims to support exploration of log data obtained from user interactions. A systematic mapping of web analytics and web usability has been reported in [14].

Regarding the visual analytics solutions existing in the IoT domain, a review of work on IoT and big data analytics is described in [18], particularly from the perspective of their utility in creating effective applications and services for several domains. In that review the authors examine the application of data analytics across IoT domains, provide a categorization of analytic approaches, and propose a layered taxonomy that defines and classifies analytics by their capabilities and application potential for research and application roadmaps.

An approach to repurposing Web analytics for the IoT is presented in [13], with the main objective of adding analytics to IoT deployments with minimal effort and cost. In that paper the authors highlight how, differently from the Web analytics domain where the overhead for incorporating analytics into a Web site is low, the IoT analytics sector is highly fragmented, more complex, with many developers that produce their own analytics dashboards often optimized for a specific application domain. Therefore, the authors highlight the benefits of leveraging existing analytics platforms and infrastructures in addressing barriers for a more widespread adoption of IoT-based analytics.

An initial attempt to provide a visual analytics tool for analysing rule-based personalization created by users has been presented in [4]. However, that tool has not been used for analysing personalization involving robots and IoT and it was only able to provide information associated with one single user per time. Therefore, it was not able to support comprehensive and summary analysis of data associated with several users, which is a key feature for more easily understanding latent trends in the datasets across various users.

3 Visual Analytics for EUD of Robotics and IoT

The main contribution of this paper is to describe a method aimed at providing suitable guidance in analysing information resulting from automated processing of datasets generated by users while interacting with an EUD tool based on trigger-action rules allowing users to specify personalisation of robot behaviour in IoT settings. The goal is to provide some means to facilitate analysis, interactive exploration and filtering of such data, so as to put users in a position to take more informed, data-driven decisions about how to improve the EUD tool and the associated applications. In parallel with this method we have also developed a prototype tool able to support the abovementioned visual analytics method.

Therefore, it is important to distinguish two levels (in terms of tools, users, tasks, etc.) involved in our approach. On the one hand, we have end users who create their personalisation rules by interacting with EUD tools. In our case, the EUD tool is called PersRobIoTE (Personalisation of Robots in IoT Environments), is based on the trigger-action paradigm, and it is aimed at supporting end users in personalising the behaviour of a robot immersed in an IoT environment. Figure 1 shows the user interface of the EUD tool when supporting the selection of relevant triggers associated with robot behaviour.

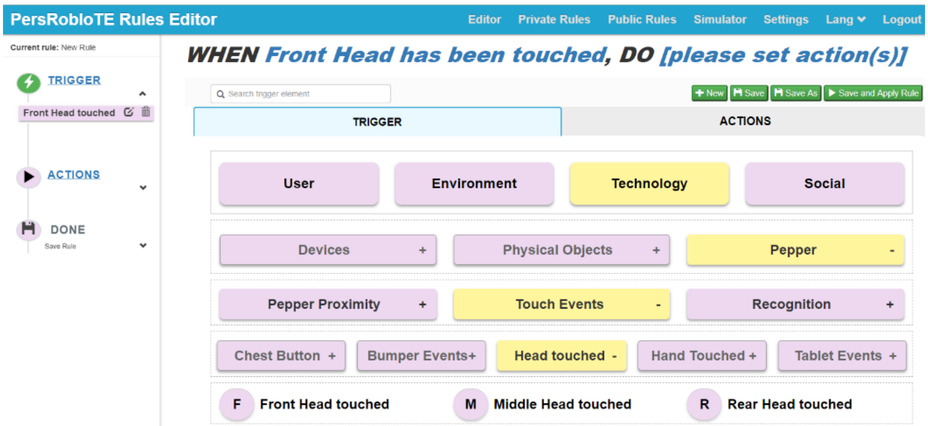

Figure 1: The EUD tool supporting the specification of robot-related triggers

On the other hand, the data associated with the interactions of users with PersRobIoTE are expected to be analysed by e.g. developers of EUD tools or IoT application developers, to understand which kind of personalisation end users are more interested in. In order to support the latter case, the visual analytics aims to provide two types of representations:

- Interactive visualisation, filtering and exploration of time-oriented events occurred during the definition of trigger-action rules (*timelines*);

- Visual representations of meaningful indicators summarising and aggregating relevant data, so as to provide users with situational awareness and monitoring (*dashboards*).

The timelines provided offer a visualization of sequences of relevant events logged during users' activities in a time-dependent manner, also allowing the user to select the most suitable time resolution/granularity to be used for the analysis. The timelines may help developers to understand whether users encounter any issues in interacting with the personalization features offered by the editors, but they provide a limited analysis about the types of rules that users were most interested in, or about the most popular trigger and action. Figure 2 shows an example of timeline-based visualization of a user session where we can see that the user created three rules in less than half an hour.

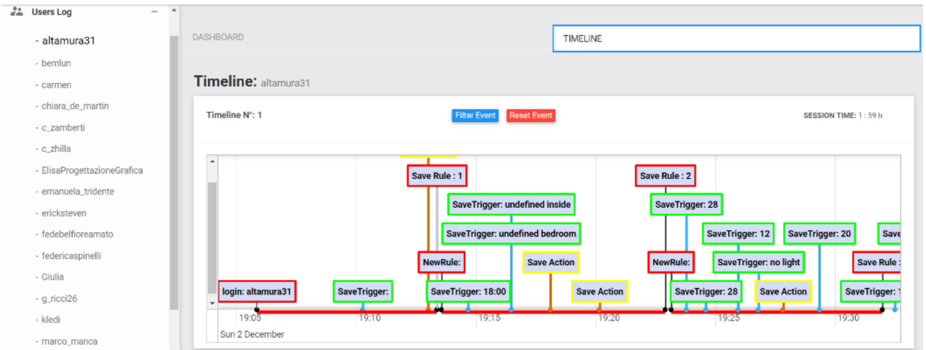

Figure 2: An example timeline

Regarding the dashboards, two main types of dashboards have been designed. One is user-related and another one is a summary, providing an overview of the data associated with multiple users. Using the first type of dashboard, as soon as a user is selected (from the list of users who interacted with the rule editor), the dashboard shows an overview of the activities carried out by the considered user: the triggers and actions created in all the sessions (where, by "session" we mean the interval of time between a user's login and a user's logout from the system), the number of rules which had been modified by the user; the rules that have been created by users described in natural language, the context dimensions (users, environment, technology, …) involved in the rules saved, the most used triggers and actions grouped by categories, the time of each interactive session, and the number of rules that have been saved and not saved in all the sessions. Indeed, since both triggers and actions are categorised into hierarchies (e.g. the high-level elements of the trigger hierarchy are the context dimensions: "User", "Environment", "Technology", "Social", and each of such elements is further refined), for each trigger/action category/dimension there is a section which shows the number of times such categories/dimensions have been used in all the defined rules. The dashboard also provides indication of how long each user session lasted overall, and also how long it took to create and save every rule.

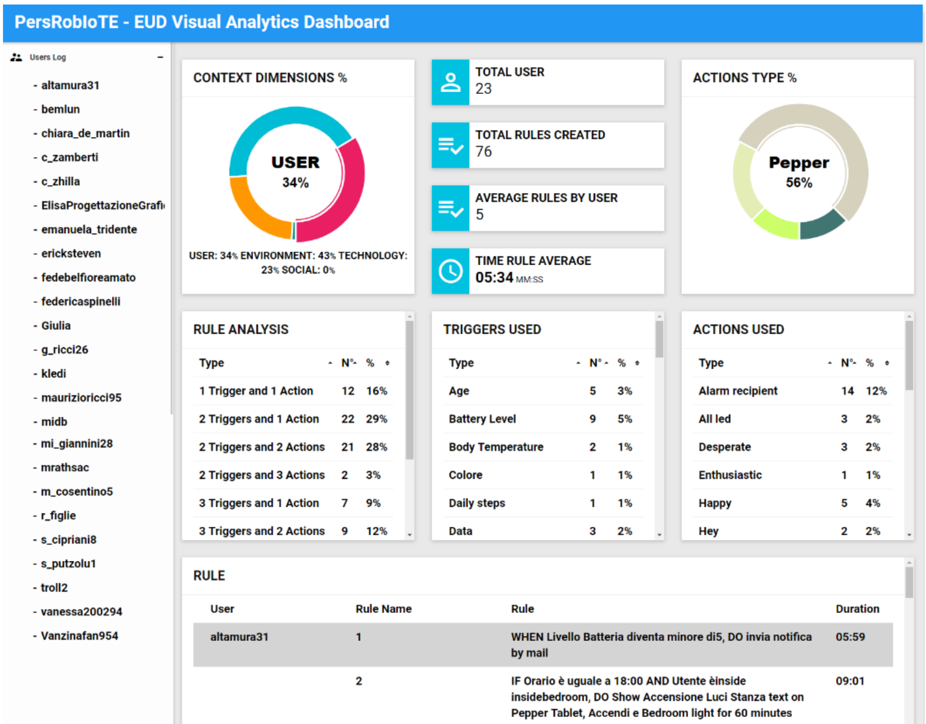

Figure 3: The Visual Analytics tool showing summary information involving user groups

As for the summary dashboard, several pieces of information are provided (see an example in Figure 3, top part of the tool):

- a pie chart showing the percentages of context dimensions exploited in the rules created by all the users;
- another pie chart showing the various action types exploited in the rules created by all the users, divided in percentages according to their types;
- some descriptive statistical data about the number of users, the total number of rules created by such users, and then the average number of rules created by each, as well as the average time spent by each in creating one rule.

Another set of data is shown in the middle part of the tool, namely:

- the various rules created by users, shown in a way that highlights their structure (in terms of number of triggers and actions involved) and how many rules have been created according to that structure (this information has been provided both in terms of absolute numbers and in percentages)
- the triggers used in the rules (absolute values and percentages)
- the actions appearing in the created rules (absolute values and percentages)

In the bottom part of the panel, the tool shows the list of the various rules created by users described in natural language.

In addition, the tool has some further interactive capabilities allowing the user to obtain further details on demand about the information presented. For instance, as soon as a specific type of trigger (or, alternatively: a type of action) is selected in one of the dashboards shown in the middle part of the tool, in its bottom part the tool highlights the set of rules that involve that trigger (or that action), so as to better inform the user about the way they have been used concretely.

4 An Example Application

4.1 The Test carried out

A test was organized in order to assess the kind of information that the analysis tool can provide when used by a number of real users. In the test we involved a class of 26 university students (12 males, 14 females) in a master's degree programme on Digital Humanities at the local University. We asked them to use the PersRobIoTE tool for creating rules involving the personalization of a robot immersed in an IoT environment. Then, we analysed the type of outputs that the visual analytics solution offers accordingly, especially those associated with the information gathered on the rules created by such users.

In the test we wanted to understand possible aspects that can be improved in the use of the EUD tool, and also assess the kind of information that the visual analytics tool can accordingly provide. The test was organized in the following manner: introduction, familiarization, test execution, questionnaire. The students were first introduced to the main functionalities of the robot and the trigger-action rule editor, which enables the definition of rules involving both the Pepper robot and a number of smart IoT things. Next, users had to create some trigger-action rules having a pre-defined structure (using the PersRobIoTE tool), and then they had to answer a set of questions collecting their feedback on various aspects of the EUD tool. After 90 minutes (including introduction and familiarization), if they did not complete the exercise, they could finish it remotely afterwards.

The tool also included an interactive module prompting users with questions to gather relevant feedback at the end of the tool use. Thus, we informed students about this, asking them to provide honest feedback, saying that the test is an evaluation of the tool and not of their performance, and that any issue they could find would be useful to improve the tool. We asked their permission to log their interactions, and informed them that the gathered data would have been used for research purposes, and we would have avoided their identification.

The tasks they had to accomplish consisted in the creation of six rules. The requested rules vary in terms of complexity, types of trigger(s) and action. In particular, they were requested to specify rules structured in the following manner:

- Rule1 (1T, 1A): Simple trigger: Technology; Simple action: Alarm
- Rule2 (2T, 2A): Composite Trigger: Environment, User; Composite action; Pepper, Light

- Rule3 (2T, 1A): Composite Trigger: Environment, Environment; Single action: Pepper
- Rule4 (2T, 2A): Composite Trigger: Environment, Technology; Composite action: Light, Pepper
- Rule5 (2T, 2A): Composite Trigger: User, Technology; Composite action: Pepper, Reminder
- Rule6 (2T, 1A): Composite Trigger: Environment, User; Single action: Pepper

Thus, the tasks submitted to users were identified in such a way to cover the various possible contextual dimensions and actions. In addition, the requested rules varied in terms of structure, i.e. both simple triggers/actions or composite triggers/actions appeared in the rules (composite actions are combined through either Parallel or Sequence operators; while composite triggers can be combined through AND, OR, NOT Boolean operators)

At the end of the use of the EUD tool, they were prompted to provide answers that were interactively shown to them, and covering the following aspects: coverage of the hierarchies of triggers and actions, clarity of the logical organization in which triggers and actions are organized, clarity of the natural language text automatically generated to describe each rule, intuitiveness of the trigger-action approach, difficulties they found during the use of the tool, possible improvements to the tool.

4.2 Analysis of the data collected through the Visual Analyzer

Using the visual analytics solution, it was possible to easily identify the most popular triggers and actions for which users wrote rules, as the tool is able to show various frequency/occurrence-related information associated with triggers and actions.
The rules created by users consisted of:

- 26 rules composed of 1 trigger and 1 action
- 52 rules composed of 2 triggers and 1 action
- 78 rules composed of 2 triggers and 2 actions

Therefore, regarding the rule structure, half the rules presented either a single trigger or a single action, whereas the remaining half consisted of 2 triggers and 2 actions.

According to the data gathered (and also as expected from the type of tasks assigned), users predominantly wrote rules for the “Environment” category (46%), “Time” being the most popular trigger type in that category, followed by “Motion”. After the “Environment” category, the triggers most used belonged to the “Technology” and also “User” categories (both scored 27%).

As expected from the exercise, the most popular actions types regard Pepper (56% of rules), “Synthesize Text” being the most used Pepper-related action (followed, in equal amounts, by “Show Text” and “Play animation”). The most frequent actions, beyond the robot-related ones, were the ones associated with “Lights” category (22%). In addition, reminders were preferred by users to be shown on tablets, followed by voice, whereas Alarms were preferred to be sent by tablet and, as a second choice, by email.

Another useful analysis that can be done through the tool is the type of rule patterns that can be identified regarding the rule structure. In this case the pattern refers to the combination of specific trigger types with specific action types. In particular, we found out that the entities under the “Environment” dimension were mostly associated with actions belonging to the “Pepper” category, followed by the “Light” category.

The entities under the Technology dimension were mostly associated with actions belonging to the “Lights” and “Alarms” categories, in equal amounts. Finally, the entities under the “User” dimension were mostly associated with actions belonging to the “Pepper” category and then, to a minor extent, to the other action categories (i.e. “Lights”, “Reminders”, “Alarms”).

It is worth pointing out that, while in this evaluation we asked students to create rules having specific structures and involving specific context types or action types, in more general cases (i.e. when users are free to create their own rules), this latter information could be key to understanding the actual personalisation preferences of users.

As for the operators for combining the triggers and the actions, the vast majority of people used the AND for combining triggers, and Sequential operator for combining actions. It is interesting to analyse the situations in which the OR operator was used to combine triggers, which can be attributed to Rule4/Task4, which requested that users combine Environment and Technology –based triggers. For instance, examples of composition of triggers involved in such rules were: “IF Living Room Light Level is low light OR LivingRoom Light Color is yellow”, “IF Time is after 23:00 OR Bed Room Light Level is no light”. This information can provide useful hints about triggers that were considered somewhat ‘equivalent’ by users (understanding when it is time to switch on some lights according either to the time, or to the light level of a room, or to the light level of a lamp).

Another set of information that the tool can provide is associated with time-related data. In particular, it is possible to know the time spent by each user to create each rule and also the average time spent in creating all the rules, as well as across the various sessions. In addition, the tool also provides the min, max, average time (and standard deviation) spent on creating rules, as calculated across all the considered users. In our case we obtained the following data: min 5 sec., max: 35 min 5 sec, average: 6 min 30 sec, st. dev.: 6 min 1 sec.

Since we have the time of day associated with the rules creation, the tool is also able to calculate when users prefer to create the rules, which in the study was between 3 pm and 6 pm.

4.3 User feedback on the Rule Editor

The 26 users were overall able to create the requested rules in a correct manner, apart from some mistakes in distinguishing between events and conditions. In the end, the 156 expected rules were created by users. After using the tool, they had to answers to a number of questions. Overall, users expressed quite positive feedback on their experience with the EUD tool. In the following, we summarize the user feedback.

- *In the rule editor, did you find all the triggers and actions you were interested in? If not, what triggers or actions would be added in the hierarchies?*

Users said that overall the most used triggers and actions were available in the hierarchies. However, the majority of them suggested further options to add in the hierarchies and regarding e.g. additional household appliances like smart washing machines, coffee machines, as well as intelligent speakers (such as Google home or Alexa), air conditioning, oven, multi-room sound systems, as well as devices for opening/closing the home shutters. Two users asked for adding further home rooms. One user asked for the possibility to have “Alarms” or “Reminders” sent by either a smart band or a smartwatch wearable device, with also the possibility to include a vibro-tactile feedback. Two users suggested adding the possibility to have Pepper interacting with the surrounding environment (e.g. turn off the tv, move objects). One user asked for better controlling the lights (e.g. dimming them within a user-specified interval of time). One user highlighted the need to have a trigger for understanding whether there is motion in all the rooms of the house (e.g. something like “Motion in house”). Another user asked for having the possibility to control in a finer manner the radio or the TV (e.g. their channels). The “Weather” trigger was also identified as a useful addition to the tool in order to have the possibility to specify rules that depend on current weather conditions.

- *Do you think the terms used in hierarchies are sufficiently comprehensible? If not, what would you change?*

The terms used in the hierarchies were judged overall understandable, except for some specific ones. For instance, one user judged the term “Hue lamp” not easily understandable for those not accustomed to (Philips) Hue lights. One user suggested providing an example for each of the terms used in the hierarchies, to improve the clarity of the tool.

- *Was it easy for you to create a mental model of what you could do? If not, where did you have any problems?*

On the one hand, four users said that, in order to create a mental model of the tool, they needed first to explore the hierarchies, in order to get an overall picture of the possibilities supported by the tool. On the other hand, one of such users emphasised that the tool was very useful in guiding the user in the process of creating rules. Another user said that the sidebar helped him a lot in following the mental models needed to build the rules. One user suggested adding some examples of rules in order to further helping users to this regard.

- *Do you think the difference between events and conditions is correctly highlighted / rendered by the instrument? If not, what would you change to improve it?*

The majority of users said that they did not find particular problems with understanding this difference. However, at the same time, they emphasised the need of further adding some hints for enabling especially people not familiar with such terms to more easily understand their different meanings. One user suggested rendering the difference between events and conditions by using some visual attributes of the associated text (e.g. using different colour or different font size or types, adding a legend). One user suggested adding this information also in the sidebar. Two users found this difference a bit difficult. However, one of them said that she, after having familiarised with the tool for a while, did not have problems anymore with such key concepts.

- *Do you have any suggestions to improve the usability / accessibility of the tool?*

One user said that further explanation and legends could be useful to better support user interactions. Another user suggested having a set of rules already available in the tool, in order to help especially novice users. Two users highlighted the need of renaming a rule without editing it from scratch. A couple of users suggested adding the possibility of having the complete hierarchy visualized at once (in the current version, apart from the top-level trigger/action entities, the other elements are refined and visualized on request, through an interactive selection). One user suggested to automatically move down the focus of the page when new elements of the hierarchy are expanded and are not visible, as in the current state explicit scrolling down is requested from the user.

- *Have you found any difficulty in expressing complex rules? If so, where?*

Building complex rules was found at a level of complexity comparable to rules composed of single triggers and single actions. Nobody complained about this aspect, one user explicitly mentioned the availability of the operators and of the rule structure in the sidebar as particularly helpful in following the specification of complex rules.

- *Did you find the feedback / error messages provided by the tool as sufficiently understandable and informative? Otherwise, in which situations?*

The majority of users said that they did not experience/notice any error message in their interactions. Regarding other feedback messages, the ones informing users of unsaved rules or concerning form fields not correctly filled in were highly appreciated by users and were found sufficiently informative.

5 Discussion

The method proposed, along with the associated tool, have shown to provide useful information in analysing the use of a EUD, trigger/action -based tool. In this case the analysis carried out enabled us to confirm that it was easy to create trigger-action rules in which even a combination of multiple triggers and multiple actions occurred, since test participants did not encounter relevant difficulties in fulfilling the tasks assigned, so producing a set of rules that were correct in their vast majority, while having just a short introduction to the approach. This to some extent motivates even more our work in proposing a method allowing people to analyse the interactions done with EUD tool, also in terms of rules produced, since it is easy to create rules, and therefore rule sets can grow very easily.

A possibility for exploiting the information that can be derived from analysing the triggers and the actions that are most used (or least used) by users could be, on the one hand, the possibility to populate the platform with examples of rules that are more frequently used by users, so as to quickly provide examples that are commonly considered as useful. On the other hand, the information associated with the types of triggers and actions that are not much used can be exploited to support users in discovering rules that involve triggers and/or actions they do not seem to be interested in, or alternatively they do not seem to be aware of.

Another useful information that can be gathered by the visual analytics solution is the kind of combinations that are most recurrent (or most preferred) by users for example in terms of complexity of the structure (e.g. 1 trigger and 1 action rather than 2 triggers and 2 actions), which can give useful insights about the typical structure in which users prefer to specify rules, and also the level of complexity according to which the rules are typically structured. Another interesting insight that such tools can provide is in the preferred combinations in terms of trigger types and action types.

In addition, the tool has some further interactive capabilities allowing users to obtain further details on demand about the information presented. For instance, as soon as a specific type of trigger (or, alternatively: a type of action) is selected in one of the dashboards shown in the middle part of the tool, the tool highlights in its bottom part the set of rules that involve that trigger (or that action), so as to better inform the user in which way they have been concretely used. Finally, it can also be useful to investigate when users feel more comfortable to specify their personalization rules.

This work can also provide contribution for strengthening the view of meta-design [22], one of the major frameworks for end-user development. Indeed, meta-design promotes the perspective of "designing the design process", in which creating the technical and social conditions for broad participation in design activities becomes as important as creating the artefact itself. The method and the tool presented in this paper are expected to improve the conditions for such collaborative design in which all the participant stakeholders incrementally acquire ownership of problems and actively contribute to their solutions. Indeed, thanks to the information provided by visual analytics tools effectively visualizing data associated with EUD tools, developers of EUD tools (aka *meta-designers*) should be able to better understand the problems of end users and then more suitably design EUD tools accordingly, so as to have end users more easily adopt EUD tools in the long term.

6 Conclusions and Future Work

In this paper we present a contribution aiming to propose the use of visual analytics to support analysis of the interactions carried out by users with trigger-action rule-based personalization tools. We have also presented the application of the proposed method to data generated by the use of the PersRobIoTE tool. However, the method proposed can be easily applied and extended to other tools with similar purposes.

Such method aims to derive higher-level information such as usage patterns followed by users while interacting with the tool, the types of rules that users were more interested in creating by using the tool, the most popular trigger and action types used, so as to facilitate the supported interactions and therefore in the end also make easy the adoption of the EUD tool and the associated applications.

We also present a first application to the data generated by a class of university students without any experience in IoT programming who were asked to use a EUD trigger-action tool for creating various rules aimed at personalizing the behaviour of a robot immersed in a IoT environment. This study offers useful material for understanding the kind of support offered by the presented visual analytics solution, and how it can be exploited and further extended.

Future work will be dedicated to adding further features to the visual analytics tool, in order to enable its users in more flexibly specifying the data they are more interested in, and it will be exploited for analysing the personalization rules created in the trials that will be carried out in an international AAL project. In addition, we also plan to carry out an evaluation of the visual analytics tool with developers and domain experts to better assess it through their feedback.

References

1. S. Alexandrova, M. Cakmak, K. Hsiao, and L. Takayama. 2014. Robot Programming by Demonstration with Interactive Action Visualizations. In Proceedings of the 2014 Robotics: Science and Systems Conference. DOI: https://doi.org/10.15607/RSS.2014.X.048
2. N. Buchina, S. Kamel and E. I. Barakova. 2016. Design and evaluation of an end-user friendly tool for robot programming. In Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN ‘16). IEEE, 185-191. DOI: https://doi.org/10.1109/ROMAN.2016.7745109
3. F Cabitza, D Fogli, R Lanzilotti, A Piccinno, Rule-based tools for the configuration of ambient intelligence systems: a comparative user study, Multimedia Tools and Applications 76 (4), 5221-5241, 2017
4. L. Corcella, M. Manca, F. Paternò, C. Santoro: A Visual Tool for Analysing IoT Trigger/Action Programming. HCSE 2018: 189-206
5. G. Desolda, C. Ardito, M Matera, Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools, ACM Transactions on Computer-Human Interaction (TOCHI) 24 (2), 12
6. Ghiani, G., Manca, M., Paternò, F., Santoro, C.: Personalization of context-dependent applications through trigger-action rules. ACM Trans. Comput.-Hum. Interact. 24(2), Article No. 14 (2017)
7. D. Glas, S. Satake, T. Kanda, N. Hagita. 2012. An interaction design framework for social robots, In Proceedings of the 2012 Robotics: Science and Systems Conference. DOI: https://doi.org/10.15607/RSS.2011.VII.014
8. N. Leonardi, M. Manca, F. Paternò, C. Santoro, Trigger-Action Programming for Personalising Humanoid Robot Behaviour. ACM Conference on Human Factors in Computing Systems (CHI'19), Glasgow, Paper 445.
9. Harms, P., Grabowski, J.: Usage-Based Automatic Detection of Usability Smells. In Human-Centered Software Engineering. Lecture Notes in Computer Science, 8742: 217-234, 2014.

10. J. Huang and M. Cakmak. 2017. Code3: A system for end to-end programming of mobile manipulator robots for novices and experts. In Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction (HRI '17). ACM, New York, NY, USA, 453-462. DOI: https://doi.org/10.1145/2909824.3020215
11. 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, May 2017
12. Xianghang Mi, Feng Qian, Ying Zhang, XiaoFeng Wang: An empirical characterization of IFTTT: ecosystem, usage, and performance. IMC 2017: 398-404
13. M. Mikusz, S. Clinch, R. Jones, M. Harding, C. Winstanley, N. Davie, "Repurposing Web Analytics to Support the IoT", IEEE COMPUTER, Sept 2015.
14. Pellizon L.H., Choma J., da Silva T.S., Guerra E., Zaina L. (2017) Software Analytics for Web Usability: A Systematic Mapping. In: Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2017. ICCSA 2017. Lecture Notes in Computer Science, vol 10409. Springer, Cham
15. E. Pot, J. Monceaux, R. Gelin, B. Maisonnier. 2009. Choregraphe: a graphical tool for humanoid robot programming. In Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN'09). IEEE, 46-51. DOI: https://doi.org/10.1109/ROMAN.2009.5326209
16. M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai. 2009. Scratch: programming for all. Commun. ACM 52, 11 (November 2009), 60-67. DOI: https://doi.org/10.1145/1592761.1592779
17. Segura, V. C. V. B., and Barbosa, S. D. J.: HistoryViewer: Instrumenting a Visual Analytics Application to Support Revisiting a Session of Interactive Data Analysis. PACMHCI 1(EICS): 11:1-11:18 (2017)
18. E. Siow, T. Tiropanis, and W. 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
19. B. Ur, E. McManus, M. Pak Yong Ho, and M. 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. DOI: https://doi.org/10.1145/2556288.2557420
20. B. Ur, M. P. Y. Ho, S. Brawner, J. Lee, S. Mennicken, N. Picard, D. Schulze, M. L. Littman: Trigger-Action Programming in the Wild: An Analysis of 200, 000 IFTTT Recipes. CHI 2016: 3227-3231
21. D. Weintrop, A. Afzal, J. Salac, P. Francis, B. Li, D. C. Shepherd, and D. Franklin. 2018. Evaluating CoBlox: A Comparative Study of Robotics Programming Environments for Adult Novices. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 366, 12 pages. DOI: https://doi.org/10.1145/3173574.3173940
22. Fischer G., Fogli D., Piccinno A. (2017) Revisiting and Broadening the Meta-Design Framework for End-User Development. In: Paternò F., Wulf V. (eds) New Perspectives in End-User Development. Springer, Cham