A Visual Tool for Analysing IoT Trigger/Action Programming

Author1
Institut
Address
State
me@me.com

Author2
Institut
Address
State
me@me.com

Author3
Institut
Address
State
me@me.com

ABSTRACT
The Trigger-Action programming paradigm has been widely adopted in the last few years, especially in the Internet of Things (IoT) domain because it allows end users without programming experience to describe how their applications should react to the many events that can occur in such very dynamic contexts. Several end user tools exist, in both research and industrial fields, which aim to support the increasing need to specify such rules. Thus, it becomes important to enrich such rule editors with functionalities able to monitor how users actually interact with them and show useful information to analyse their use. In this short paper, we present a visual tool for monitoring and analysing the use of a trigger-action rule editor. The goal is to provide a tool useful to better understand what the end users' personalization needs are, how they are expressed, how users actually carry out the specification of rules, and whether users encounter any issues in interacting with the personalization features offered by the editors. The proposed solution supports the analysis through a set of timelines describing the actual use of the personalization tool, with the possibility to select specific events of interest, and provides data useful for understanding the types of triggers, actions and rules actually composed by users, and whether the personalization features offered by the editors are effectively exploited by users.

CCS CONCEPTS
• **Human-centered computing** → **Visual analytics**; • **Human-centered computing** → User interface programming

KEYWORDS
Trigger Action Programming, Visual Analytics, Log User Interaction

1 INTRODUCTION
The environments where we live and act are increasingly characterised by the presence of a multitude of interactive devices and smart objects interconnected with each other. Since we interact with our applications in such very dynamic and unpredictable environments it is not possible to foresee at design time how an application should react to all the possible contextual changes that can occur during its use.

For such reasons in order to obtain applications able to adapt to the context of use in an effective way it becomes important to allow end users themselves to 'program' the behaviour of their applications.

In this trend trigger-action programming has emerged as a useful and intuitive approach. Users can personalise the application behaviour through sets of rules indicating triggers and consequent effects. Triggers can be either instantaneous events (corresponding to context changes) or conditions that, when satisfied, activate the execution of specific actions. This type of approach has stimulated several contributions both from research [3,5,8] and industrial viewpoints (IFTTT, Tasker, Resonance AI, …). In particular, IFTTT has been particularly successful. It has more than 320,000 automation scripts (called "applets") offered by more than 400 service providers. The applets have been installed more than 20 million times, and more than half of IFTTT services are IoT devices related [9]. Thus, we can foresee in the near future an increasing interest to environments allowing people to provide many rules to personalize their context-dependent applications. In this perspective, having some tools able to analyse how actually users try to personalise their context-dependent applications with such approaches can become very useful not only for developers of trigger-action authoring environments, but also for application developers and domain experts. However, differently from other existing tools that exploit log analysis for usability evaluation purposes, in this case the goal is not strictly to understand whether there is some bad user interface design, but rather to see what the personalization needs are, how they are expressed by users, and whether they have some conceptual problems in expressing them.

2 RELATED WORK
Both in research and industrial fields there has been interest in the trigger-action programming to allow users to define their own adaptation rules. From the commercial point of view IFTTT[1] is one of the most used application. It provides mechanisms to create rules composed of one trigger and one action. Triggers are events occurring in the user's context, and which cause the execution of an action. They are grouped according to their intended goal, i.e. environment control & monitoring, calendars & scheduling, news & information. Resonance AI[2] is a tool for developers which aims to automate and personalize every application. It provides contextual awareness services to enhance products and services with real-time understanding and reactions based on the current

[1] https://ifttt.com

[2] https://www.resonance-ai.com

user's environment. Such data become actionable triggers that developers can use to automate or suggest actions, allowing to personalize apps and devices behaviour.

From the research perspective, we started our study from TARE [5] a Trigger-Action Rule Editor that provides the possibility to create rules more flexible than IFTTT since they can be created as a composition of multiple triggers and actions. In this area, Desolda et al. [3] developed EFESTO, a visual environment that allows users to express rules for controlling smart objects. The followed paradigm is based on a model, called 5W, which defines some specification constructs (Which, What, When, Where, Why) to build rules coupling multiple events and conditions exposed by smart objects, and for defining temporal and spatial constraints on rule activation and actions execution. Coutaz and al. presented AppsGate [2], an EUD environment designed to empower people with tools to augment and control their home. AppsGate aims to support different activities such as monitoring the home state and programming its behaviour in a context-dependent manner. Another similar approach is [4] ImAtHome, an IOS application built over Apple HomeKit, and allowing home inhabitants without any programming skills to control home automation by means of creating scenes and rules for defining the complex behaviour of a smart home. Still in this area Metaxas and Markopoulos [8] defined the concept of affinity regrouping heuristics and presents the mechanisms applied throughout the contextual ranges of the involved services. The semantic information that the services disclose lets the editor recognize this affinity and allows it to group them together in logical expressions since they refer to the same aspect (e.g. user's activity). The context-range editor supports end users formulate logical expression regarding the context. The proposed monitoring and visualization method can be useful to analyse the user interactions with such tools as well, since they still support trigger-action rules for IoT applications.

One typical use of the information contained in logs of user interactions is for usability studies. Palmer [10] presents different metrics for measuring usability, and lists different types of methods to evaluate a user interface. UsaProxy [1] exploits a proxy-based solution to access remote web pages: the proxy adds some JavaScript code to specify the listeners which log the user interaction with the concerned page(s). The output produced by the proxy is a simple list representing the IP address of the connected device, the visited pages, and some events' description, without any particular visualization able to support their analyses. MUSE [11] also exploits a proxy server in order to insert in the target web pages some code to log user interactions. The logged events are shown in a timeline representation in which it is possible to compare a timeline representing the 'optimal' interaction with the one expressing the 'real' user interactions in order to help designers to discover some usability issues in the user interaction. WELFIT [12] is a tool to identify usage patterns based on client-side event logs and by presenting event stream composition characteristics. The system records usage data during real use, identifies usage patterns, and indicates potential user interface design problems. Harms and Grabowsky [6] proposed to transform 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. Differently from such proposals, in this work we focus on providing designers of trigger-action rule editors with interactive visualisations supporting exploration and filtering of the logged relevant interaction data, so as to derive higher-level information such as the types of rules that users were interested in creating with the tool, the most popular trigger and action types used, and the types of usage patterns followed by users while interacting with the tool. Moreover, by analysing the users' log it is possible to understand what the personalization needs are, how they are expressed by the users, whether their rules actually support the desired results, if the personalization features offered by the editors are sufficient.

3 THE SUPPORTED ANALYSIS

In this study we considered the TARE tool (Trigger Action Rule Editor) [5], which allows users to define their trigger action rules in an intuitive way. The tool is flexible in the order in which users specify the rules (they can start either from triggers or from actions), they can re-use a previously defined rule in order to create a new one. Moreover, they can combine multiple triggers by using the Boolean operators AND and OR.

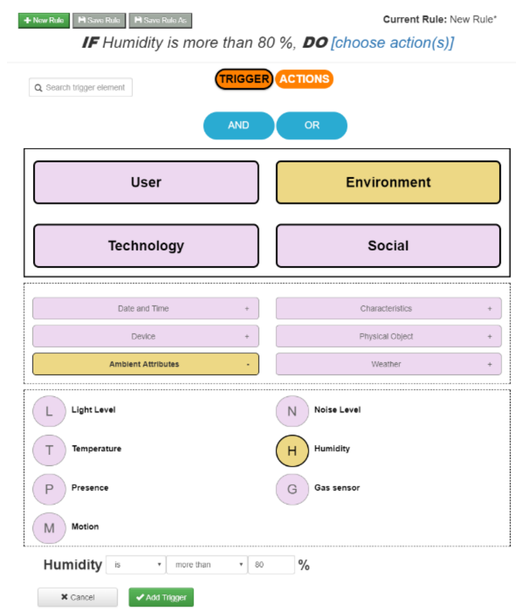

Figure 1: TARE editor

Figure 1 shows how users navigate through the context hierarchy by selecting the context dimension to which the considered trigger belongs and by traversing the *context categories* (and sub-categories), which logically group together related context elements. The leaves of the context hierarchy refer to *context entities*, and are used to specify triggers. For instance, in Figure 1 the trigger is "IF humidity is > 80%, and it involves the context entity "humidity". The same hierarchical structure is used to categorize the actions. In addition, TARE provides users with the possibility to search for a specific trigger, by specifying a concept to search for in the hierarchy of triggers. TARE has been validated in three different trials that involved 58 users in total. During the tests users had to perform various tasks concerning the specification of personalization rules with different complexity in terms of number of triggers and actions. Thus, by observing the results in the trials and considering previous work in the area, we identified some features that an environment supporting the analysis of how users define context-dependent personalization rules should provide to facilitate such analysis.

One important requirement of a visual analyser is the availability of interactive data exploration: the tool should provide users with different zooming levels, as well as the possibility to select individual items and get specific details on demand. In addition, it should also provide different interactive features to enable users to focus on different aspects of trigger-action programming. In the case of the type of tool considered in this study, relevant information includes: the most recurring/frequent context entities used in rules, the most recurring combinations of trigger types and action types used in created rules, the most recurring sequences of usage patterns logged, etc. For this purpose, in order to analyse the behaviour of users interacting with TARE, we identified a number of meaningful events to log, thus excluding some low-level events to log (like e.g. mouseover, mouseout, blur) which were judged not particularly relevant for the type of planned analysis. The logging implementation was done by a JavaScript file which appends handlers to various events supported in TARE and related to rule creation, editing, saving. In particular, we found useful to log user's selections of:

- "New Rule", "Save Rule", "Save Rule as", "Edit Rule" and "Delete Rule" buttons (used to manage rules);
- "Triggers" and "Actions" buttons (used to go to the part of the tool dedicated respectively to trigger and to action specification);
- "AND" and "OR" buttons (used to compose two triggers);
- Trigger/Context Dimension (User, Environment, Technology, Social), to select one specific trigger dimension;
- Action Dimension Selection (Update/Distribute UI, Change Appliance State, Activate Functionalities, Alarm, Reminder), to select one specific action dimension;
- Trigger type, to select a specific type of trigger within the hierarchy of triggers;
- Action type, to select a specific type of actions within the hierarchy of actions;
- Trigger Operator (e.g. equal, different, more, less; to select the operator involved in the trigger specification);
- Action Operator (e.g. turn on-off, open, close; to select a specific type of action);
- Event/Condition (to specify whether the statement involved in the trigger specification refers to an event or to a condition);
- Insert specific Trigger value;
- Save/Update/Cancel Trigger or Action;
- Search Trigger Element.

By analysing the logs produced during user sessions, the tool provides the possibility to know, for each user and also across users, the following information:

- From which rule part users prefer to start editing the rules (whether first from triggers and then the actions or vice versa);
- The sequence of trigger/action dimensions and entities that users have passed through to reach the trigger/action leaf;
- The time spent to create a rule (max, min, average);
- The time spent to edit a rule (max, min, average);
- The number of rules created in each session and in all sessions;
- The number of triggers/actions created in each session and in all sessions;
- The number of rules/triggers/actions that users started to edit without saving them.

We also judge it useful to provide designers with the possibility to filter the results listed above to allow the user to configure the list of events of interest on which they want to focus, as well as to provide further quantitative information such as the context dimension and the trigger entity that have been most used in defined triggers. A similar type of analysis has also been provided for actions. Such summarised representations of the users' sessions are particularly useful when the number of events becomes very high and difficult to manage.

For each trigger/action entity it is possible to identify the optimal path to follow for reaching it. For instance, in order to reach *Respiration Rate* entity a user has to select the *User* dimension and then traverse the subcategories *Physical and Mental* and *Physical*. In this way the tool can compare the real path performed by the users with the optimal one, which can be useful to understand the difficulties that users faced during the trigger definition. In addition, we can also compare the performance of two users performing the same tasks, and to derive information about their interaction patterns.

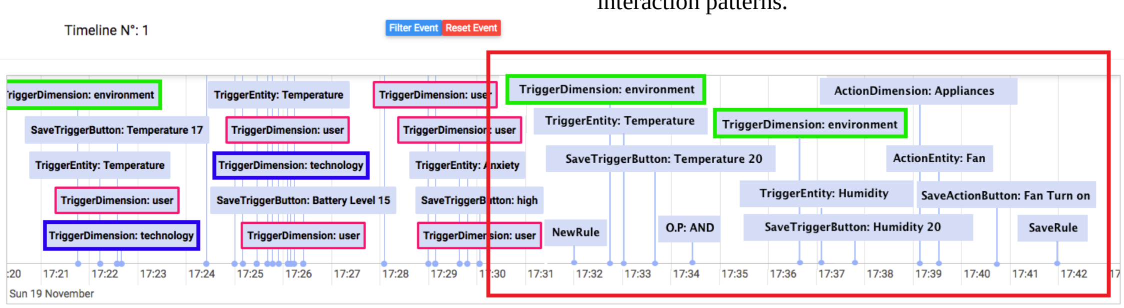

Figure 2:Example of Timeline Representing Events during Rule Composition

4 THE TOOL VISUALIZATIONS

In order to visualize in a simple manner the sequence of events logged during users' activities we decided to use a dynamic timeline visualization [7] (see Figure 2), in order to provide a time-dependent overview of the relevant events occurred. For each user the tool shows a set of timelines, each one presenting the list of events recorded in an interactive session. On the X axis the timeline shows the sequence of relevant events ordered by time, also indicating the time when the events occurred. However, due to the large amount of events that can be recorded, it may happen that the timeline could be difficult to interpret. For this reason, we provide users with a functionality that allows them to filter events according to specific criteria. For instance, it is possible to select the events that involve a specific context dimension. In addition, in order to allow users to better perceive the differences between the various types of events, users can select a specific colour to assign to each of them. Figure 2 shows an example timeline: in this case it is possible to see that at the beginning the user interacts with the hierarchy of events by following a quite "explorative" pattern, selecting different context dimensions (e.g. first environment, then user, then technology) and some elements in their lower levels in a 'sparse' manner i.e. without focusing on particular contextual entities to use for specifying the intended rule. As such, this part of the timeline can be interpreted as reflecting the behaviour of a user who wants to familiarise with the event hierarchy and the included concepts. In the second part of the timeline (highlighted by the red rectangle) the behaviour of the user is more focused, the interactions are more locally concentrated on e.g. a few events/actions belonging to a specific category (i.e. environment), and the user in the end is also able to finalise the rule editing by saving the rule. These two different usage patterns in the first part and in the second part of the timeline can be identified by the fact that each trigger dimension/action type is associated with a different colour and then, in the first part of the timeline several different colours appear, whereas in the second part of the timeline there is just one prevailing colour (the one associated with the dimension finally used in the rule). One interesting type of analysis that can be done concerns the use of the search functionality included in the rule editor. For example, a large use of the search functionality by a user can suggest that this user found quicker to use that functionality instead of exploring the trigger top-down hierarchy. Furthermore, it could be useful to compare what users specify in the "search" field and what they actually used afterwards in the rule specification; or analysing whether, after searching for an element, users actually found an element of interest and proceeded with the rule specification or they needed to repeat the process multiple times (and which terms they used in such repetitions). In addition, when a specific trigger element that was already used in a rule is repetitively searched by the same user using the search functionality (instead of exploring the hierarchy) could be a sign that the position of that concept within the hierarchy is not very logical for that user and therefore not easy to locate.

Another related source of information provided is a dashboard (Figure 3) with an overview of the activities carried out by the considered user: the rules that have been specified (described in natural language), the context dimensions involved in the rule editing (see the pie chart on the right), the most used triggers and actions grouped by dimensions and the time of each working session. In addition, the tool shows information about the number of rules that have been completed and not completed (see the bar charts visualised in the bottom part and clustered by session). Such bars are interactive and the user can select each bar to get the details of the concerned rules.

Figure 3: The dashboard presenting summary information.

4 CONCLUSIONS

In this short paper we present a method and the features of a supporting tool for analysing the users' behaviour when interacting with a trigger-action rule editor for personalising their context-dependent applications. We discuss the more relevant features for this analysis and provide example visualizations that can be provided.

While in this work we applied the approach to a specific tool (TARE), the type of analysis of the users' behaviour presented can be easily extended to other similar tools supporting trigger-action programming of IoT context-dependent applications.

Future work will be dedicated to empirical validation of the visual tool proposed.

REFERENCES

[1] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. 2006. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In Proceedings of the 15th international conference on World Wide Web (WWW '06). ACM, New York, NY, USA, 203-212. DOI=http://dx.doi.org/10.1145/1135777.1135811

[2] Joelle Coutaz and James L. Crowley. 2016. A First-Person Experience with End-User Development for Smart Homes. IEEE Pervasive Computing 15, 2 (April 2016), 26-39. DOI=http://dx.doi.org/10.1109/MPRV.2016.24

[3] Giuseppe Desolda, Carmelo Ardito, Maristella Matera: End-user development for the internet of things: EFESTO and the 5W composition paradigm. In: Daniel, F., Gaedke, M. (eds.) RMC 2016. CCIS, vol. 696, pp. 74–93. Springer, Cham (2017). doi:10.1007/978-3-319-53174-8_5

[4] Daniela Fogli, Matteo Peroni, Claudia Stefini: ImAtHome: Making trigger-action programming easy and fun. J. Vis. Lang. Comput. 42: 60-75 (2017)

[5] 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

[6] Patrick Harms and Jens Grabowski. 2014. Usage-Based Automatic Detection of Usability Smells. In Proceedings of the 5th IFIP WG 13.2 International Conference on Human-Centered Software Engineering - Volume 8742 (HCSE 2014), Stefan Sauer, Cristian Bogdan, Peter Forbrig, Regina Bernhaupt, and Marco Winckler (Eds.), Vol. 8742. Springer-Verlag New York, Inc., New York, NY, USA, 217-234. DOI=http://dx.doi.org/10.1007/978-3-662-44811-3_13

[7] David M. Hilbert and David F. Redmiles. 2000. Extracting usability information from user interface events. ACM Comput. Surv. 32, 4 (December 2000), 384-421. DOI=http://dx.doi.org/10.1145/371578.371593

[8] Georgios Metaxas and Panos Markopoulos. 2017. Natural Contextual Reasoning for End Users. ACM Trans. Comput.-Hum. Interact. 24, 2, Article 13 (April 2017), 36 pages. DOI: https://doi.org/10.1145/3057860

[9] Xianghang Mi, Feng Qian, Ying Zhang, XiaoFeng Wang: An empirical characterization of IFTTT: ecosystem, usage, and performance. IMC 2017: 398-404

[10] Jonathan W. Palmer. 2002. Web Site Usability, Design, and Performance Metrics. Info. Sys. Research 13, 2 (June 2002), 151-167. DOI=http://dx.doi.org/10.1287/isre.13.2.151.88

[11] Fabio Paternò, Antonio G. Schiavone, and Pierpaolo Pitardi. 2016. Timelines for Mobile Web Usability Evaluation. In Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '16), P. Buono, R. Lanzilotti, and M. Matera (Eds.). ACM, New York, NY, USA, 88-91.

[12] Vagner. F. de Santana and M. C. Calani Baranauskas "WELFIT: A Remote Evaluation Tool for Identifying Web Usage Patterns through Client-Side Logging" in International Journal of Human-Computer Studies, vol. 76 no. C pp 40-49, 2015.