

Recommendations for Personalization Rules in Block Composer

Andrea Mattioli^a, and Fabio Paternò^a

^a CNR-ISTI, HIIS Laboratory, Via Moruzzi 1, 56124 Pisa, Italy

Abstract

This paper reports about a possible approach to integrating a recommendation system (RS) for giving support to users during the creation of Internet of Things (IoT) personalization rules on the Block Composer tailoring environment. Recommending for end-user creation of rules requires a specific way to model the problem different from a user-item matrix filling approach. The items to be recommended (trigger-action rules) are not unitary objects but are composed of several parts, assembled in a step-by-step process. This feature makes recommendations for this end-user development (EUD) approach similar to the sequence-based recommender systems and can be exploited to provide more suitable suggestions.

Keywords

Recommender systems, personalization rules, sequence-based recommendations

1. Introduction

In recent years several solutions have been put forward to supporting end users in creating personalization rules in IoT applications according to the trigger-action programming approach. In this area, we have developed a system for the end-user definition of trigger-action rules, Block Composer [6], which provides an original use of the puzzle metaphor to support users in creating such rules. The editor implements a “free” composition style, and the user is supported in the rule creation process by colour and shape coherence of blocks, which suggest the correct placement of the rule elements, and by a text area to explain problems identified in the defined rule. Further support is provided by a Recommendation System (RS), through the presentation of recommended relevant rule examples in a secondary workspace (see figure 1), basing on the input provided by the user during the rule creation. Figure 1 shows an example of a rule under development, where only the trigger is specified, and on the lateral panel there is a suggestion of a possible relevant rule.



Figure 1: Snapshot of a section of the tailoring environment user interface, which shows on the left the work area with a rule on editing, and on the right the secondary workspace with a suggested rule.

A first instantiation of the system was tested with users without experience in programming, who reported to appreciate the RS support, and to have used it both as a method to check the correctness of the created rule and to discovery new functionalities of the devices. However, only half of the participants reported to have used the RS when it was not explicitly requested by the tasks proposed in

EMPATHY: Empowering People in Dealing with Internet of Things Ecosystem. Workshop co-located with AVI 2020, Island of Ischia, Italy.

EMAIL: andrea.mattioli@isti.cnr.it (A. Mattioli); fabio.paterno@isti.cnr.it (F. Paternò)

ORCID: 0000-0001-6766-7916 (A. Mattioli); 0000-0001-8355-6909 (F. Paternò)



© 2020 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

the user test. More proactive suggestions focused on the next step to take in order to complete a rule was reported as a desired feature.

For this reason, we propose a different method to generate recommendations in this context. As discussed in [11], rule recommendation for the IoT brings new challenges: the definition of “users” and “items” in this context is not immediate. Rules, the “items” to suggest, are not unitary entities, but are composed of triggers, actions, and operators between them. Also, the composition of a rule is a compound process made of several steps. Therefore, suggestion of a completely defined rule selected in a rules repository is an approach that may not be completely relevant to support users during rule composition.

A more appropriate way to look at the suggestions in this context is in terms of a process. The idea in this case is that the user can receive, at each step of the process, an appropriate suggestion about the next step to take basing on what has done and user profile information. In the EUD for IoT context, this can be obtained through a framework which generates recommendations each time the user selects a rule element (trigger or action) from the toolbox, which represents a step in the sequence that needs to be performed. A list of top suggestions will be provided to the user, who can select one of these elements (providing an implicit feedback on the quality of the suggestion) to automatically add it to the rule in construction. A session can be interpreted as the sequence of tasks that should be performed by the user from the beginning of the editing until the final step (save or discard the created rule).

Following the classifications in [8], such a RS can be defined as a session-aware (because we know the previous interactions of the user with the system) sequence RS taking into account the structure of the trigger-action rules (because the task is to generate suggestions related to the current point in the process, and the elements should be inserted in meaningful way).

2. Related Work

Regarding RS in EUD and IoT some initial solutions have been presented. [4] adopts a description of the rules using a semantic graph approach, which abstracts from the technological details of devices (brands, manufactures) enriched with collaborative information about user preferences and feedbacks. This structure is used to train a model that predicts the relationship between users and rules. [5] presents a recommendation framework for the IoT context, based on device usage statistics integrated with context data. A further segmentation framework is also presented in order to generate recommendations thanks to common usage patterns amongst users living in similar contexts. In [11] a recommendation method which considers user-item-device interactions is introduced, recognising the need to take the rules’ device requirements and user’s device into account. The system also considers dynamic implicit feedback regarding the user’s interest in a rule.

An application of sequence-based RS is [9], where a factored Markov Decision Process (fMDP) was applied to detect the topic of a user navigation via clickstream analysis. The problem of the too wide state space of the fMDP was mitigated assuming the independence of the attributes of the items. An approach for the e-commerce context is [10], where recommendations are generated using only data from the current sessions, joined with the item features. In [7] a multi-step RS for a cooking app was presented, to provide suggestions at the actual point in the execution of the recipe, making use of past and future data (because the next steps in a recipe are known). The RS is a multi-decision classifier, which at each step of the cooking sequence decides if present or not a recommendation, its type and format, on which device display it.

Related applications of RSs in a EUD perspective can be found in the Web mashup context. In [2], a semantic approach to identify pattern candidates is used in conjunction with a collaborative filter to rank the candidates basing on their popularity, ratings and relevance respect to the current user. In [3], a procedure to integrate an automated mashup generation tool with a pattern recommendation system

was developed. The former is based on auto-deriving a composition that can satisfy the user goals, elicited via a dialog-based interface. Via the obtained information, an overlay model of the domain ontology is generated, and used by a SPARQL-based query system to retrieve widgets of interest for the user. The latter system uses information from existing workspace models generated by users, reacts on user actions during composition querying for recommendations, then filters and ranks them basing on the current composition context.

3. Suggestion Pipeline

Basing on the relevant aspects identified in the related work we propose an approach to recommendation which can be subdivided in a three-stage pipeline.

The first stage is the pre-filtering of rules not relevant to the actual user/situation. For example, only the rules of the neighbours of the current user should be considered or should have more weight. Also, the rule goal could be used if known. To find the most similar users, the similarity between created rules and/or between sensors activation patterns (as in [1, 5]) could be used. The latter approach may require some extra steps to ensure the privacy of the users' data.

The second stage is the building of a model for the sequence recommendation. Filtered rules have to be deconstructed into their basic constituent elements. These rule fragments can be a device/service, with its related features (the associated value, the distinction between event and condition type trigger if it is specified, the operator which connects to the next rule element, such as "and", an operator applied to the same element like the negation, the position in the composition sequence), and then used to train a model that will sort the candidates basing on their utility in the context. Different models can be used for this task: graph-based (for example using knowledge graphs), Markov Chains-based approaches, machine learning sequence prediction models or hybrid models. Basing on the adopted model, different data can be used to evaluate the utility of a rule fragment. Other than rule element and its features, user profiles should also be used in the model building. A basic approach to get a "next step" in a sequence can be to directly assign a weight to the transitions, obtained by how many times a certain transition occurs in the rules dataset, and sorting the suggestions using this metric. However, this approach is short-sighted, because it only considers the strength of a single link, discarding the influence of the previous ones, and does not try to optimize a long-term reward. Also, it only considers the popularity factor.

Another aspect to be considered is the interpretability of the model. Graph or Markov-based approaches can represent rules as sequences of transitions from a rule element to another, each one with a pointer to the next element in the rule sequence. This transitions list can be directly translated in a visualization that can make the reasoning process of the model more easily represented and understood, making them particularly suited to provide users with explainable recommendations. Other models can be interpreted using model-agnostic techniques, such as partial dependence plot. For more advanced or hybrid model provide an interpretable visualization may be more difficult.

The third stage regards the instantiation of the suggestion in the actual context. This means checking if the local IoT installation includes objects which can perform the functionalities of the suggested rule elements (for example using a lightweight semantic approach), and the relationships of the suggestions with the already inserted rule elements and with rules previously defined by the user, that can be positive (exploration of more behaviours of an item or completing a behaviour made of more rules) or negative (conflict with other rules). To comply with this, both the element type (e.g. room temperature) and the element value (e.g. $> 20^{\circ}\text{C}$) should be considered and weighted with user long-term goals (e.g. comfort or energy saving). Also, the direct user intervention can be used at this stage to further refine the suggestion list, for example using the selecting or discarding of the suggestion to model a contextual bandit over different rule editing moments, like rule just started, trigger part completed, rule ready to be saved.

4. Conclusions and future work

In our future work, we plan to better define and instantiate the presented method in a new RS for Block Composer. An important aspect to address is the definition of a method for the second stage of recommendation, which is how to build a model capable to assign weights to the transitions considering the past experiences. As reported in [8], sequence modelling for session-based RS is mainly based on Markov Models, Reinforcement Learning and Recurrent Neural Networks. Reinforcement learning seems a promising approach because it is naturally sequence-based and works via interactions with the environment. Even a not full application of reinforcement learning (e.g. standard or contextual bandit) can be of use in this context, to apply a refinement to the suggestion list. With this approach some aspects of the user preferences can be implicitly evinced from the user's interaction with the system (e.g. rule construction style, preferences for a specific device or room, changes of preferences over time). However, user's long-term preferences and goals can be used at all recommendation stages, and it have to be considered how to model them to provide more tailored suggestions.

5. Acknowledgements

This work has been supported by the PRIN 2017 "EMPATHY: Empowering People in Dealing with Internet of Things Ecosystems", www.empathy-project.eu/.

6. References

- [1] Barbara Barricelli and Stefano Valtolina. 2017. A visual language and interactive system for end-user development of internet of things ecosystems. *Journal of Visual Languages & Computing*. 40.
- [2] Radeck Carsten, Alexander Lorz, Gregor Blichmann, and Klaus Meißner. Hybrid recommendation of composition knowledge for end user development of mashups. *ICIW 12 (2012)*: 30-33.
- [3] Soudip Roy Chowdhury, Olexiy Chudnovskyy, Matthias Niederhausen, Stefan Pietschmann, Paul Sharples, Florian Daniel, and Martin Gaedke. 2013. Complementary assistance mechanisms for end user mashup composition. In *Proceedings of the 22nd International Conference on World Wide Web (WWW '13 Companion)*. <https://doi.org/10.1145/2487788.2487919>
- [4] Fulvio Corno, Luigi De Russis, and Alberto Roffarello. 2019. RecRules: Recommending IF-THEN Rules for End-User Development. *ACM Transactions on Intelligent Systems and Technology 10 (09 2019)*, 1–27. <https://doi.org/10.1145/3344211>
- [5] Hanjo Jeong, Byeonghwa Park, Minwoo Park, Ki-Bong Kim, and Kiseok Choi. 2019. Big data and rule-based recommendation system in Internet of Things. *Cluster Computing 22, 1 (01 Jan 2019)*, 1837–1846. <https://doi.org/10.1007/s10586-017-1078-y>
- [6] Andrea Mattioli and Fabio Paternò. 2020. A Visual Environment for End-User Creation of IoT Customization Rules with Recommendation Support. In *International Conference on Advanced Visual Interfaces (AVI '20)*. <https://doi.org/10.1145/3399715.3399833>
- [7] Elnaz Nouri, Robert Sim, Adam Fourney, and Ryen W. White. 2020. Step-wise Recommendation for Complex Task Support. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval (CHIIR '20)*. <https://doi.org/10.1145/3343413.3377964>
- [8] Massimo Quadrona, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Comput. Surv.* 51, 4, Article 66 (September 2018), 36 pages. <https://doi.org/10.1145/3190616>
- [9] Maryam Tavakol and Ulf Brefeld. 2014. Factored MDPs for detecting topics of user sessions. In *Proceedings of the 8th ACM Conference on Recommender systems (RecSys '14)*. <https://doi.org/10.1145/2645710.2645739>
- [10] Bartłomiej Twardowski. 2016. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. <https://doi.org/10.1145/2959100.2959162>

- [11] Beidou Wang, Xin Guo, Martin Ester, Ziyu Guan, Bandeeep Singh, Yu Zhu, Jiajun Bu, and Deng Cai. 2018. Device-Aware Rule Recommendation for the Internet of Things. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18). <https://doi.org/10.1145/3269206.3272009>