POLITECNICO DI TORINO
Repository ISTITUZIONALE

End User Development in the IoT: a Semantic Approach

(Article begins on next page)

End User Development in the IoT: a Semantic Approach

Alberto Monge Roffarello
Politecnico di Torino
Corso Duca degli Abruzzi, 24 Torino, Italy 10129
Email: alberto.monge@polito.it

***Abstract*—The Internet of Things (IoT) is, nowadays, a well recognized paradigm. In this field, End User Development (EUD) is a promising approach that allows users to program their devices and services. The representation models adopted by contemporary EUD interfaces, however, are often highly technology-dependent, and the interaction between users and the IoT ecosystem is put to a hard test. The goal of my research is to explore new approaches and tools for helping end-users to program their technological devices and services. For this purpose, I proposed EUPont, an ontological model able to represent abstract and technology independent trigger-action rules, that can be adapted to different contextual situations. EUPont has been evaluated in terms of understandability, completeness, and usefulness. Currently, I am using the semantic features of the model in different research projects, e.g., to optimize the layout of EUD interfaces, and to design a recommender system of trigger-action rules. Preliminary results are promising, and confirm the benefit of using the semantic information of EUPont for helping end-users to better deal with the forthcoming IoT world.**

***Index Terms*—Internet of Things; End-User Development; Trigger-Action Programming; Semantic Web**

I. Introduction

The potential of the Internet of Things (IoT) is nowadays being increasingly recognized. The IoT can be seen as a set of different items that include not only physical devices, which are likely to increase dramatically over the next years [1], but also on-line services such as social networks or news portals. Although the expected benefits brought by the IoT are known (e.g., sharing objects, services, and data), the advent of such massively interconnected items raises new challenges, especially concerning how people interact with the complex IoT ecosystem. As reported by Cerf and Senges [2], many IoT products have a level of complexity that prevents users from investing time and resources in learning how to configure them. In addition, contemporary IoT systems adopt an app-centric or device-centric approach towards their users, as they are too often designed with an industry-centered approach that promotes vertical silos [3]. For these reasons, direct interoperability between most contemporary IoT devices and services is not possible *by construction*. Nowadays, end user personalization of IoT devices and online services can be mainly obtained through third party cloud-centric solutions such as IFTTT[1] and Zapier[2]. Such End-User Development (EUD) interfaces empower users to define joint behaviors between sets of IoT devices and online services by means of trigger-action rules, e.g., “if the NEST camera in the kitchen detects a movement, then turn the Philips Hue lamp in the kitchen on.” Despite apparent simplicity, the composition of such rules is not an easy task for end-users. The representation models adopted by contemporary EUD solutions, in fact, are often highly technology-dependent, and are not suitable to face with the rapid growth of the available interconnected “things”. In the case of IFTTT, for example, such technology dependency forces users to be aware of the manufacturers and models (or service brand) of each involved item. Thus, contemporary EUD solution often expose too much functionality, and become too complex for non-programmers. Consequently, end-users without technical skills may not find these systems useful [4]. The following scenario better explains the problems that end-users may encounter:

John is always hot, especially in summer. He loves air conditioning, and he would like to set a low temperature wherever it is possible. At home, John has an intelligent Nest thermostat that he controls through his Android smartphone. John goes to work by car. There, all the offices are equipped with a Samsung smart air conditioner.

John must define many different rules to personalize his IoT ecosystem, at least one rule for the home, one for the car, and one for the office, even if the rules perform the same logical operation, i.e., set a specific temperature when he enters a place. Furthermore, with such a low level of abstraction, he must know all the involved devices and services before composing any rule, e.g., *Nest*, *Samsung*, *Android*. In other words, similar rules do not adapt to different contexts, and John cannot define any personalization without knowing all the technological details. To take a step towards a higher level of abstraction, I designed EUPont [5], an EUD ontological model for the IoT. EUPont allows the modelling of abstract and technology-independent trigger-action rules that can be adapted to different contextual situations. With EUPont, users can define rules based on their final functionality. John, for example, is now able to define a single rule for his need, e.g., *“if I enter any defined locations, then set the temperature to 20 Celsius degree.”* EUPont is available on the web[3], and it

[1] *http://ifttt.com* (last visited on December 5, 2017)
[2] *http://www.zapier.com* (last visited on December 5, 2017)
[3] *http://elite.polito.it/ontologies/eupont.owl* (last visited on December 5, 2017)

has been evaluated in terms of in terms of understandability, usefulness, and completeness.

Currently, I am investigating two applications of the model to further explore my research goal, i.e., new approaches and tools for helping end-users to program their IoT devices and online services. The goal of the first one, named EUDOptimizer, is to employ combinatorial optimization methods to enhance EUD interfaces. By using models of human performance and layout perception, along with the semantic information provided by EUPont, EUDoptimizer reorganizes the elements presented to the users in such interfaces in an optimal way. In a second research project, named RecRules, I developed a recommender system of trigger-action rules for the IoT. In this case, rather than helping people to compose trigger-action rules, the goal of RecRules is to directly suggest them. By using the semantic information of EUPont, RecRules is able to suggest rules on the basis of their final functionality. Results of preliminary evaluations of the projects are promising, and confirm the potential of the research approach: the usage of the semantic information defined EUPont for supporting users to personalize their devices and services could help them to better deal with the forthcoming IoT world.

II. Background and Related Works

End-User Development (EUD) empowers users to define and tailor joint behaviors between IoT devices and online services in various areas, like the home, the car, or for a healthy lifestyle. Nowadays, with the continuous growth of the IoT ecosystem, people are increasingly moving from passive consumers to active producers of information, data, and software [3]: in the last 10 years, several commercial tools for end user personalization of IoT devices and services, such as IFTTT or Zapier, are born. The connection between recent and past solutions is the underlying programming approach. Trigger-action rules are one of the most popular approaches and offer a very simple and easy to learn solution for creating end user IoT applications, according to Barricelli and Valtolina [6].

As the number of available interconnected "things" grows, however, our relationship with such objects changes and new user needs and novel challenges emerge [2]. Some previous works tried to mitigate the complexity of the contemporary IoT ecosystem by acting on the underlying models, with the aim of simplifying the trigger-action rules composition process. Barricelli and Valtolina [6] presented an extension of the trigger-action paradigm to better cope with the evolving IoT scenario, by incorporating other IoT users, space and time, and the social dimension. In a test with more than 200 participants, Ur et al. [7] discovered that end-users express triggers in one level of abstraction higher than the one adopted by contemporary EUD environments, e.g., they say "when I am in the room" instead of "when motion is detected by the motion sensor." By following the need of taking a step towards a higher level of abstraction in the EUD, I designed EUPont [5], an End-User Development ontological model for the IoT. With EUPont, users can express triggers and actions in terms of their final functionality: the goal is to replace the contemporary device-centric approach with a more user-centered interaction, where users express their needs without worrying about any technological details.

Besides the adopted representation models, other works focused on new interfaces and tools for EUD [8], [9]. Desolda et al. [8], for example, reported the results of a study to identify possible visual paradigms to compose trigger-action rules, and presented the architecture of a platform to support rules execution. In a similar way, Ghiani et al. [9] presented a method and a set of tools to allow end users without programming experience to customize the context-dependent behavior of their Web applications through the specification of trigger-action rules. In line with such recent works, I am investigating new approaches and tools for helping end-users to program their IoT devices and online services. What differentiates my research is the usage of semantic technologies. Adding semantics to the IoT is a topic of particular interest: the lack of open IoT standards naturally leads to a semantic-oriented perspective, and researchers agree that the IoT could benefit from a semantic approach in terms of interoperability, data integration, and knowledge extraction [10].

III. EUPont: End User Programming Ontology

EUPont [5] aims at defining a high-level representation for end user development in the IoT. The goal is to allow the definition of abstract and technology-independent trigger-action rules, and to support their execution, thus allowing run-time environments to adapt them to different contextual situations. Figure 1 shows the general architecture of EUPont.

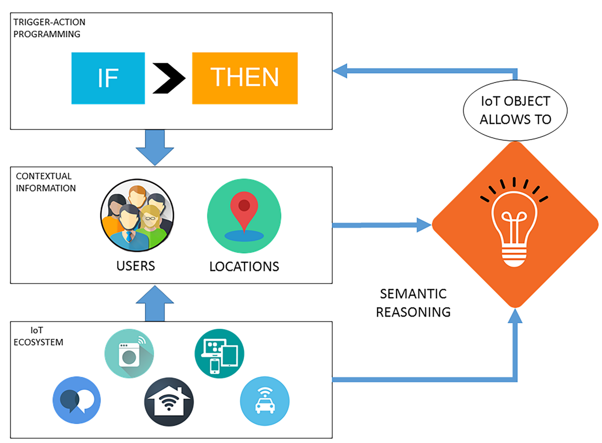

Fig. 1. The architecture of EUPont.

The model is composed of three main layers:

- **Trigger-Action Programming (TAP)**: a conceptualization of the trigger-action programming approach. It defines a hierarchy of triggers and actions to be used for defining rules in the trigger-action form. Triggers and actions are hierarchical organized in such a way that the user can decide her preferred level of abstraction. With the Trigger-Action Programming layer, users can generally ask to lighten a room, for example, or they can be more specific, e.g., by referring to the room lights, or to a specific light bulb.

- **Contextual Information (CI)**: the concepts describing locations and users that act as attributes for triggers and actions, and define the contextual information of the IoT ecosystem.
- **IoT Ecosystem (IoT)**: an abstract representation of IoT devices and online services. Devices and services are classified on the basis of their categories (e.g., lighting systems, user devices, smart appliances) and their final capabilities (e.g., switching, sensing, actuating, communication).

Beside the definition of trigger-action rules (TAP), the model supports their real-time execution. By reasoning on the IoT and CI layer, in fact, EUPont is able to map the defined trigger-action rules with devices and services in the IoT Ecosystem that are able to reproduce the desired abstract behaviors, by taking into account the current context, dynamically. EUPont has been evaluated in a user study [11] with 10 participants related to the composition of trigger-action rules. In particular, the model has been compared with the representation model used by IFTTT. Results shows that EUPont was well understood by end-users and it was suitable for creating trigger-action rules. In fact, the new representation improved the correctness of the rules composed by the participants, and, as expected, it allowed the participants to compose the rules in less time. In a further evaluation [5], the expressiveness of EUPont has been evaluated by presenting an automated translation procedure of 290,963 IFTTT rules collected in a large dataset by Ur et al. [12]. Results show that EUPont is as least as expressive than the representation model used by IFTTT, and it is fully compatible with IFTTT rules. Moreover, the flexible trigger-action approach adopted by EUPont increases the expressiveness of the representation.

IV. EUPont in Practice

I used the EUPont model in two ongoing research projects. The fist approach, named EUDoptimizer, aims at defining an optimization method to automatically generate optimal EUD interfaces, while the second one, named RecRules, aims at defining a recommender system of IoT trigger-action rules.

A. *EUDoptimizer*

In the most popular contemporary EUD interfaces, e.g., IFTTT and Zapier, the modality for composing trigger-action rules is the same. For defining a trigger (or an action) users must first select from a grid menu an IoT device or an online service, on which they can define a particular trigger (or action) and the relative details. Despite the rapid growth of the IoT ecosystem, however, IoT devices and online services are not presented in a meaningful order to the user. Through a web-scraping process on IFTTT, for example, we found that they are simply ordered by their internal identifier, thus making it difficult to find the desired one (in IFTTT there are, as of today, more than 400 available IoT devices and services). With EUDoptimizer, I defined a combinatorial optimization method to enhance EUD interfaces. The goal is to reduce the time effort needed by end-users to compose trigger-action rules. For this purpose, EUDoptimizer defines a multi-objective task to order IoT devices and online services according to their usage probability, while maintaining logical groups of related elements. The idea is to move towards the top of the menu the most frequently used objects, while grouping similar objects to help end-users in finding the desired element. In the multi-objective task, I exploited a state-of-the-art predictive model of user performance in menu search, named Search-Decision-Pointing (SDP) [13], in combination with two different models of item groupings to take into account *semantic* and *functional* similarities between IoT objects. With the *semantic* model, based on the EUPont ontology, the system tries to keep together objects that belong to the same categories and application areas, e.g., home appliances, social networks, etc. Instead, with the *functional* model, the system uses EUPont to find similarities between the offered triggers and actions in terms of the final behaviors they aim to define.

EUDoptimizer has been integrated in IFTTT. Figure 2 shows a screenshot of the interface automatically generated by exploiting a Simulated Annealing algorithm. We can observe that the 10 IoT devices and online services most frequently used (according to the dataset of Ur et al.) are prominently placed in the 10 positions closer to the top of the menu. Furthermore, IoT devices and online services with semantic and functional similarities are pulled together in logical groups. In the right side of Figure 2, for example, there are many items related to photos and videos (*iOSPhoto, Android Photo, Eyefy, Youtube, Flickr, Dailymotion, and 500px*).

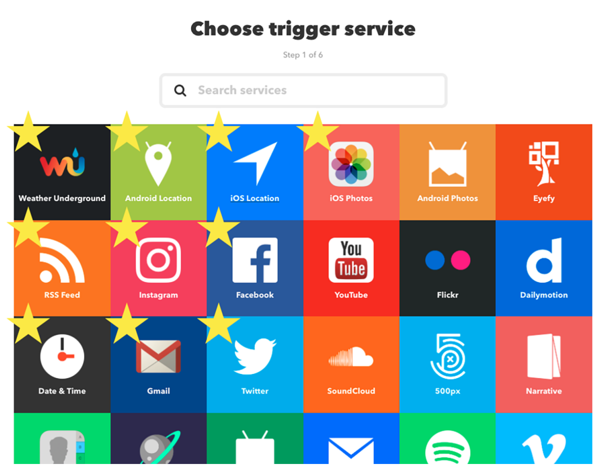

Fig. 2. An example of an optimized menu layout. The stars indicate the 10 most frequently used IoT objects according to the dataset of Ur et al.

EUDoptimizer has been preliminarly compared with IFTTT in a user study with 12 participants. The goal was to understand whether the optimized interface improved the user performances in the composition of trigger-action rules with respect to the “normal” version. Results show that trigger-action rules were composed in less time by the participants with the optimized interface. Even for very uncommon rules, for which IoT devices and online services were not placed on the top of the menus, EUDoptimizer partially reduced the time effort needed by participants to complete the tasks. This tells us that, with the continuous growth of the IoT ecosystem, an optimized EUD interface could reduce time and cognitive effort needed by end-users for selecting items between hundreds of devices and services.

B. RecRules

In the RecRules project, I am investigating a different approach. In this case, rather than helping people to compose trigger-action rules, the goal of RecRules is to directly suggest them to be activated. With the low-level of abstraction offered by the contemporary EUD interfaces, in fact, the number of possible combinations between different devices and services is very high, and the number of shared rules is growing. This particular type of information overload could be addressed through recommender systems: recommendation techniques could make a meaningful contribution towards increasing the usability, the acceptance, and the popularity of End-User Development [14].

RecRules is a hybrid and semantic recommendation algorithm that addresses semantic information and collaborative user preferences in a graph-based setting. To compute recommendations, RecRules first translates trigger-action rules into the EUPont representation. Then, it extracts ontological information, builds a collaborative semantic graph, and exploits path-based features to train a learning to rank algorithm. By exploiting the semantic information of EUPont, the algorithm is able to suggest trigger-action rules on the basis of their final functionality, and it can be exploited to compute recommendations for yet unknown technologies, thus helping users to discover new IoT devices and online services. RecRules has been evaluated in terms of accuracy of the top-N recommendations on real data extracted from IFTTT [12]. Preliminary results show that the accuracy of the recommendations increases by considering the semantic information of EUPont, and the algorithm is able to compute similar recommendations even for devices and services not yet used by the user, according their final functionality.

V. Conclusions and Future Developments

In this paper, I presented my reasearch on End-User Development. In this field, contemporary soultions such as IFTTT and Zapier offer highly technology-dependent representation models, and are not suitable to face with the rapid growth of the number of available interconnected "things". The starting point of my work is EUPont, a high-level semantic model I designed for defining abstract and technology-independent trigger-action rules, that can be adapted to different contextual situations. Based on EUPont, I am currently investigating two further approaches for helping people to customize their technological devices and services: an optimization method to automatically generate EUD interfaces (EUDoptimizer), and a recommender system of trigger-action rules (RecRules).

Overall, preliminary results are promising, and confirm the potential of the research approach: the usage of semantic technologies, and, in particular, of EUPont, could help end-users to better deal with the forthcoming IoT world. Future works will include new evaluations of the proposed approaches. For EUDoptimizer, for example, the multi-objective function could be further analyzed: do end-users prefer semantic or functional similarity between IoT devices and online services? Which of the two impact more end-user efforts? How do icon images and colors influence the selection of IoT objects? In RecRules, instead, a more user-cenetered evaluation is needed to confirm the obtained accuracy results.

References

[1] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," Cisco Internet Business Solutions Group, Tech. Rep., 2011.

[2] V. Cerf and M. Senges, "Taking the Internet to the Next Physical Level," *IEEE Computer*, vol. 49, no. 2, pp. 80–86, Feb 2016.

[3] D. Munjin, "User Empowerment in the Internet of Things," Ph.D. dissertation, Université de Genève, May 2013. [Online]. Available: http://archive-ouverte.unige.ch/unige:28951

[4] T.-H. K. Huang, A. Azaria, and J. P. Bigham, "Instructablecrowd: Creating if-then rules via conversations with the crowd," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '16. New York, NY, USA: ACM, 2016, pp. 1555–1562.

[5] F. Corno, L. De Russis, and A. Monge Roffarello, "A semantic web approach to simplifying trigger-action programming in the iot," *Computer*, vol. 50, no. 11, pp. 18–24, November 2017.

[6] B. R. Barricelli and S. Valtolina, *End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings*. Cham, Germany: Springer International Publishing, 2015, ch. Designing for End-User Development in the Internet of Things, pp. 9–24.

[7] B. Ur, E. McManus, M. Pak Yong Ho, and M. L. Littman, "Practical trigger-action programming in the smart home," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 803–812. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557420

[8] G. Desolda, C. Ardito, and M. Matera, "Empowering end users to customize their smart environments: Model, composition paradigms, and domain-specific tools," *ACM Transaction on Computer-Human Interaction (TOCHI)*, vol. 24, no. 2, pp. 12:1–12:52, Apr. 2017.

[9] G. Ghiani, M. Manca, F. Paternò, and C. Santoro, "Personalization of context-dependent applications through trigger-action rules," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 24, no. 2, pp. 14:1–14:33, Apr. 2017.

[10] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the internet of things: Early progress and back to the future," *International Journal on Semantic Web and Information Systems*, vol. 8, no. 1, pp. 1–21, January 2012.

[11] F. Corno, L. De Russis, and A. Monge Roffarello, "A high-level approach towards end user development in the iot," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '17. New York, NY, USA: ACM, 2017, pp. 1546–1552.

[12] B. Ur, M. Pak Yong Ho, S. Brawner, J. Lee, S. Mennicken, N. Picard, D. Schulze, and M. L. Littman, "Trigger-action programming in the wild: An analysis of 200,000 ifttt recipes," in *Proceedings of the 34rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: ACM, 2016, pp. 3227–3231.

[13] A. Cockburn, C. Gutwin, and S. Greenberg, "A predictive model of menu performance," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 627–636.

[14] W. Haines, M. Gervasio, A. Spaulding, and B. Peintner, "Recommendations for end-user development," in *Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI)*, 2010.