Fabio Paternò, Gabriella Lucci (2015):
User Conceptual Models of Event-Action IoT Applications.
In International Reports on Socio-Informatics (IRSI),
Proceedings of the CHI 2015 - Workshop on End User Development in the Internet of Things Era
(Vol. 12, Iss. 2, pp. 75-80)

User Conceptual Models of Event-Action IoT Applications

Fabio Paternò, Gabriella Lucci

CNR-ISTI, HIIS Laboratory
fabio.paterno@isti.cnr.it

Abstract. In this position paper we discuss why the event action model is important for IoT applications. We report on an analysis about how it is currently supported in some Android apps, the usability and expressiveness of such solutions, and provide indications for the design of new EUD environments for IoT applications able to support it through smartphones.

1 Introduction

In recent years we have witnessed the use of computers in an increasing number of dynamic contexts, with a huge variety of users having radically different backgrounds and a plethora of diverse tasks to perform. On the one hand, such users share a common requirement for software, to support their common tasks that may vary rapidly, with some of them that cannot be anticipated at design time but discovered only during actual use. On the other hand, slow software development cycles and lack of domain knowledge by software developers are limitations to addressing the requirements of different users. End-User Development (EUD) [1] can help to mitigate this gap. However, another emerging trend is the Internet of Things (IoT) era [2], where ‘smart’ physical objects are thought as networked together, able to interact and communicate with each other, with human beings and/or with the environment to exchange data and information ‘sensed' about the environment, reacting autonomously to events in the real/physical world, and influencing it by running processes that trigger actions and perform services. According to Gartner[1], there will be nearly 26 billion devices on the Internet of Things by 2020: in this scenario IoT applications need to address deeply contextualized user needs.

So far main EUD approaches [3] have mainly considered the desktop platform and applications that are unable to control smart objects, and adapt to the changing context of use [4]: desktop spreadsheets have been the most used EUD tools so far. Thus, there is a need for a new generation of EUD approaches able to address IoT ubiquitous applications [5] and to offer unprecedented opportunities to achieve deeper, more meaningful and faster personalization by putting the user at the centre of informative systems, ambient and personal sensors, communicative tools, and mobile and ubiquitous computing devices.

In particular, our aim is to reach a better understanding of the user mental models when they want to specify how the interactive application should behave according to the objects available in the context of use. A first contribution in this direction [6] has only considered some aspects related to practical trigger-action programming in the smart home by using the IFTTT ("If This Then That") Web environment. Such environment has limited possibilities, since it only supports applications with only one trigger and one action. Thus, we need environment able to allow users to specify more flexible behaviours that can be triggered by various combination of events, and activate multiple effects.

2 An Analysis of Three Apps for EUD in Smartphones

We started our work with an analysis of how three Android Apps (Tasker[2], Locale[3], and Atooma[4]) aim to support non-professional developers to create context-dependent applications by exploiting the smartphones' sensors and capabilities [7]. They provide three different solutions according to the event / condition / action model. They have similar structures: categories, elements, actions. We have conducted an analysis of the three environments from two viewpoints: expressiveness (to what extent they support the relevant concepts); usability (for which a user study has been carried out).

---

1 http://www.gartner.com/newsroom/id/2636073

2 http://tasker.dinglisch.net

3 http://www.twofortyfouram.com

4 https://play.google.com/store/apps/details?id=com.atooma&hl=it

They use slightly different vocabularies for the same concepts. In Atooma an application is called Atooma and is structured in an IF and a DO part. Locale supports the development of situations described in terms of Conditions and Settings. Tasker is used to create Profiles structured into Contexts and associated Tasks composed of Actions. In terms of the development process, Atooma is completely sequential: developers have first to indicate the conditions and then the actions. In Locale it is not sequential and developers can freely choose to specify situations and settings in any order. Tasker is semi-sequential in the sense that the starting point can be either the condition or the action, but if a condition is specified then the corresponding actions must be indicated. The three environments differ in terms of how they implement the events and actions model. Right at the beginning Atooma asks users to select mainly from four main macrocategories, Locale provides a list of elements, which can be extended through plugins, while Tasker structures the selectable events and conditions in terms of six Contexts.

Tasker has the greatest expressiveness (more than double Locale's), with a number of actions that can be expressed (108) greater than the triggers (83). In Atooma the number of expressible conditions (70) is greater than the actions (48). In both triggers and actions, Locale has the same number of expressible elements (40) and is the one that has the lowest total expressiveness. On a total of 80 features, 58 are obtained through plugins since few elements are directly integrated into the environment.

In the user test, two tasks were accomplished in ascending difficulty order for each app. The first was to specify an adaptation rule composed of one event and one action, the second included an event, a condition and two actions. In the first task, with Locale the mean time of the first task execution (1'29") was about half than Atooma (2'52"), Tasker (3'50") was almost a minute longer than Atooma. In the second task, the application that required less average execution time was again Locale (3'35'') followed by Atooma (4'00'') and Tasker (5'14'').

To summarize, the most expressive environment (Tasker) is also the one that was found most difficult to use (as demonstrated by the highest performance time, error numbers, and unsuccessful performance numbers).

3 Card Sorting for Identifying User Conceptual Mode

In order to better understanding how users classify the concepts that characterize context-dependent applications we carried out a user study by using card sorting and associated cluster analysis techniques. The identification of the cards derived from the analysis the three apps. We proposed cards able to indicate all the events and actions that were supported by the three apps. In the end we obtained 39 cards: 14 referring to only events, 6 to actions, and 19 were used for both events and actions. The card sorting was proposed to 18 users. At the beginning we provided them with some basic concepts to introduce context-dependent applications, then the users had to group logically them and assign a name to each group identified. They had to carry out the exercise twice: once to classify the 33 cards related to events and once for the 25 cards representing the possible actions. During the exercise the groups identified by the users were entered in the UXsort tool that has been used to support the results analysis. By applying hierarchical clustering methods the tool is able to measure the linkage among elements groups and produces a dendrogram that represents the similarity among elements through a tree-like structure. The supports their analysis by using three clustering algorithms (single linkage, complete linkage, average linkage).

In order to select the most interesting results we decided to focus on solutions that offer a number of groups between 5 and 8. Such numbers were identified by the analysis of the numbers of groups supported by current solutions, and also considering that with less than five groups we can obtain groups containing heterogeneous elements, and with more than 8 groups the solutions tend to separate elements that people would expect together. This generated 12 solutions for grouping the events and 12 for the actions.

By observing the pattern elements in the groupings occurring both in the events and actions classification we have identified the associations between elements and groups that users found more meaningful, and the corresponding group names that were assigned more frequently occurring. We also discarded solutions with an unbalanced number of elements in the resulting groups or with a group name not completely consistent with the actual elements (e.g archive for a group containing the media-player element).

In the end, we found a solutions with two small variants that differ for the collocation of the GPS element. In one variant, the GPS element was moved from the Connection to the Sensors group and this caused the creation of one further group for the actions. In the other variant, there is one further group

(Localization) for both events and actions that contains the GPS and Location elements. A more detailed description of this study is reported in [8].

4 Conclusion and Future Work

We have presented a study aiming to identify how users classify most recurrent events and actions in IoT context-dependent applications. The study has also considered how such issue has been addressed in current solutions for Android devices. Future work will be dedicated to designing a new environment for smartphone able to take into account user mental models and support the possibility of specifying adaptation rules through an expressive language, such as the AAL-DL (Advanced Adaptation Logic Description Language) [9] in a platform for context-dependent adaptation of IoT applications.

5 References

[1] H. Lieberman, F. Paternò, M. Klann, V. Wulf, "End-user development: An emerging paradigm" In: Lieberman, Henry, Paternò, Fabio and Wulf, Volker (eds.). "End-user development (Human-Computer Interaction Series)", pp. 1-8 Springer, 2006.

[2] L Atzori, A Iera, G Morabito, The internet of things: A survey, Computer networks 54 (15), 2010, pp. 2787-2805

[3] A. J. Ko, et al.. 2011. The state of the art in end-user software engineering. ACM Comput. Surv. 43, 3, Article 21 (April 2011), 44 pages

[4] F. Paternò (2013) End User Development: Survey of an Emerging Field for Empowering People, ISRN Software Engineering, vol. 2013, Article ID 532659, 11 pages, 2013.

[5] J. Danado, F. Paternò: Puzzle: A mobile application development environment using a jigsaw metaphor. J. Vis. Lang. Comput. 25(4): 297-315 (2014)

[6] B. Ur, E. McManus, M. P. Y. Ho, M. L. Littman: Practical trigger-action programming in the smart home. CHI 2014: 803-812

[7] G. Lucci, F. Paternò: Understanding End-User Development of Context-Dependent Applications in Smartphones. HCSE 2014: 182-198

[8] G. Lucci, F. Paternò, Providing Users with Intuitive and Expressive Models of Contextual Event-Action Behaviours, Submitted Paper

[9] F. Paternò, C. Santoro, D. Spano, Serenoa EU Project, AAL-DL: Semantics, Syntaxes and Stylistics, 2013.