End-User Development for Lifelogging and eWellness

Stefano Valtolina
Università degli Studi di Milano
Via Comelico, 39/41
20135 Milan, Italy
valtolina@di.unimi.it

Barbara Rita Barricelli
Università degli Studi di Milano
Via Comelico, 39/41
20135 Milan, Italy
barricelli@di.unimi.it

AVI, June 07–10, 2016, Bari, Italy

Abstract

The growing popularity of Internet of Things' devices, sensors, and applications available on the market is pushing towards a revolution of lifelogging and its relationship with wellness. Thanks to the characteristics of pervasiveness and mobility of wearable IoT tools, we are witnessing the emergence of a new research and development domain: the eWellness. Together with this, new challenges arise, and ask for effective methods and techniques to improve the way extraction, merge, analysis, visualization, and data sharing are currently managed. In this paper, we present our research study on the eWellness domain and in particular the design and development of the SmartFit framework, aimed at supporting the collaborative visual design of IoT rules by means of End-User Development techniques.

Author Keywords

End User Development, Internet of Things, Design, Human Factors; ETL (extract, transform and load) operations.

ACM Classification Keywords

H.3.4 [**Systems and Software**]: Distributed systems, Information networks; H.4 [**Information Systems Applications**]: Miscellaneous; H.5.2 [**User Interfaces**]: GUI, Interaction styles.

Introduction

Nowadays we are witnesses of the proliferations of different sensor devices able to produce several types of data that can be profitable used for advising people on how to improve their lifestyle, health, and wellness. The survey reported in [1] underlines that many tools in the Internet of Things (IoT) field, mainly focus on hardware devices (sensors and actuators) and on artificial intelligence techniques that may help to add sophisticated capabilities to the processing of information provided by hardware devices. A current challenge in this domain concerns the need to conceive it as a socio-technical system, which encompasses not only a variety of hardware and software components, such as sensors, actuators, mobile apps and Web applications, but also people that are bound together by social ties and personal relations and that are virtually linked each other. This leads to what is to be considered the Internet of Things ecosystem. In this context, the main interest is to provide the users with tools able at properly helping them in making sense of the data originated from different sources (i.e., devices, sensors) and for enabling them to understand and exploit huge amount of data and events that are eventually collected in time. In our research, we mainly study applications of IoT technologies designed for a specific scenario: The eWellness domain in which people can collect data gathered through several devices for monitoring and keeping track of weight, sport/fitness activity, calories intake, and sleep quality. In this domain, what is currently missing is a solution able to support domain users (experts in monitoring or in supervising physical activity of individual or teams) in selecting relevant data sources and compose them through End-User Development (EUD) techniques [2], so that users are enabled to self-construct their contents to respond to their situational needs. In this scenario, end users find themselves at the center of a complex scenario that they need to manage in efficient, effective, and satisfactory way. EUD represents the ideal approach for empowering end users and make them becoming unwitting developers in their own environment [3,4,5].

A scalable and efficient solutions should be devised that can be applied by domain experts in configuring the network of sensors (both physical and social), and services. ETL (extract, transform and load) solutions, usually applied off-line, need to be revised and applied on-line for feeding the system with fresh and timely data. Finally, a user-friendly environment has to be conceived for properly supporting the user in the analysis processes (sensor discovery, feeding, and knowledge inference).

Specifically, in this paper we present and discuss our sociotechnical study on the design and development of IoT ecosystems to be used by non-professional sport teams. The peculiar structure of non-professional sport organizations is characterized by the existence of small teams with athletes who live different kind of lives, being professionals in different domains and meeting only for some hours a week. Keeping track of their habits, in terms of physical activity, nutrition, sleep and so on, would help the coaches in understanding the variety of the team members and finding successful schemes of training. In order to enable domain experts in generating, configuring and managing the network of sensors and services we adopt a meta-design approach. By means of this approach we seek to create social-technical conditions for broad participation in design activities at both design and use time of all the involved stakeholders, rather than anticipating all requirements at design time [6]. Objective, techniques and processes are observed to empower users to be active designers, creating their own situated solutions that can be translated in a configuration of sensors and services.

The SmartFit Case Study

In this section, we present a use case we studied to experiment the potentials of our solutions both at meta-design and design level. In this context of use we consider three different Communities of Practice (CoPs) [7] that co-exist in a eWellness scenario: IoT engineers, Coaches and Trainers, and Athletes. IoT sensors/devices engineers are in charge of connecting, maintaining, and setting up the devices and sensors to be used by coaches, trainers, and athletes. Their system named StremaLoader, allows the design of data flows by aggregating data sources and applying operators to them for merging, joining and transforming the gathered data. More details about this environment are referred in [8,9]. The StreamLoader output is a flow of events that can be used by coaches and trainers for defining rules to monitor particular situations and to adopt suitable actions according to the occurrence of given conditions. Exploiting a rule language, the Rule Editor enables coaches and trainers to act as end user developers by designing the ECA statements to be used to supervise athletes' performances and lifestyle and they also analyze the gathered data in their interactive system. Finally, athletes can be seen as the real end users, the ones who passively use the data gathered by the IoT sensors and devices to better their lifestyle and sport performances. A tailored interactive system can be used to have a view on their behavior and performances at any time during the day.

Meta-Design Phase: The StreamLoader

At meta-design level, IoT Engineers need to configure the network of sensors and services for managing the data-flow to be served at the Design level interactive system. The outcome of this environment is to detect a set of events that coaches and trainers need to manipulate at design level for monitoring the physical activities or daily behavior of their athletes. StreamLoader provides IoT Engineers with a set of ETL operators used for manipulating streams of data in real time (or almost real time). In this way, data are loaded from a set of sensors and services, providing a convenient way for user to real-timely read the data information and make tactical decisions. Commercial systems such as Talend Studio[1], StreamBase Studio[2], Waylay.io[3] offer graphical interfaces for designing workflows and dataflows as graphs of connected nodes representing tasks and data-sources. While Talend works on static data coming from fixed data-sources, StreamBase and WayLay can receive and analyze continuous data streams and are specifically designed for IoT. These environments provide rich user interface support for the full application lifecycle but they are desktop-based systems and in some cases, specific conditions can be only created by adopting strategies based on programming languages paradigms (as for StreamSQL in StreamBase Studio) or by personalizing existing templates having well-defined trigger policy (as in Waylay.io). Visual queries are other examples of solutions can be used for assisting users to extract information from databases in an intuitive, visual and

---

[1] www.talend.com

[2] www.streambase.com

[3] www.waylay.io

natural approach, making information systems comprehensive and efficient for a wide range of applications. SmartVortex Visual Query System [10] is a typical application where the visual strategy of querying takes the form of drawings or graph. Also in this case the expected users have limited competence for managing data-streams but they wish to easily express basic queries with little efforts and no competence in developing code. Starting from the study of current solutions based on graphical visual strategies for executing ETL operations, we designed a meta-design environment for helping the IoT engineers to manipulate events. In our approach, one of the most important EUD activity takes place during the meta-design of the flow of events that characterize a context of use. For example, in a gym, IoT engineers need to combine data coming from different sensors and devices for properly defining the exercises. These EUD activities are exploited for executing efficiently and effectively Event ETL operators. So far, several operations have been developed for processing and combining the streams produced by the sensors [8]. In the fitness example, the events might concern the user’s physical condition, which will be used for suggesting what exercises are better to perform, or which precautions to follow for improving the quality of live (for example, by changing the diet plan). In our context, the meta-design environment is a Web environment where IoT engineers can drag and drop different sensor data sources and visually apply on them a set of operations. This application offers an engine and graphical environment for data transformation and mashup. As depicted in Figure 1 (A), this mash-up consists of a user interface that contextually displays icons of data sources or operations in order to link, aggregate, merge or transform data coming from different sensors. It relies on the idea of providing a visual workflow generator for letting the domain experts to create combination of data originated by sources. In Figure 1, the IoT engineer has taken data coming from an application used for annotating the calories intake and three sensors: Two for counting steps, and one for monitoring the sleep. The engineer has then used a merge operator for merging data coming from the two step-counters and a transform operator for transforming the minutes of sleep in hours. Finally, s/he has joint these data into a flow that has been then collected into the data-warehouse. An advanced use of such visual paradigm allows the users to have an online generation of sample data coming from the data sources dragged-and-dropped on the canvas, or as result of the operations carried out on them (see Figure 1 (B)). Following this strategy, information is gathered from the network of devices and specific operations can be triggered on user requests.

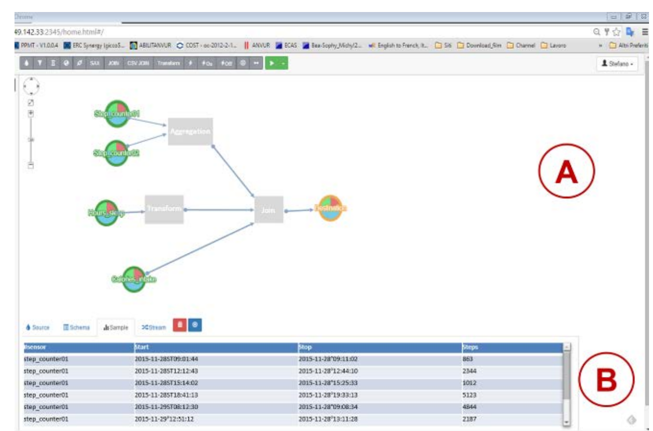

Figure 1. A screenshot of the StreamLoader

Design Phase: Rule Editor

The meta-design environment described earlier in this Section and used by IoT Engineers, constitutes the base for providing coaches and trainers with the possibility to manipulate the flow of events in order to monitor the physical activities or daily behavior of their athletes. The design environment for manipulating these events aims at offering a graphical visual strategy for exploiting the potentials of an IoT environment in the deployment of rules. Systems like JBoss Drools[4], OpenRules, and IBM WebSphereJRules[5] provide platforms for supporting users in creating complex rules to trigger proper actions when specific conditions occur. In these environments, the coding of complex rules is generally performed by skilled technicians through ad-hoc Rule Engines. Nevertheless, in real contexts of use, the definition of rules is performed by domain experts that are not experienced technician and prefer to use natural language or graphical notations. Other visual strategies typically used in IoT field for modelling ECA (Event-Condition-Action) rules can be described through the most famous systems that apply them: IFTTT, Atooma and Yahoo's Pipes. The first two examples allow users to define sets of desired behaviors in response to specific events. The visual strategy aims at creating automated rules by using graphical notation for programming statements such as: "IF this DO That" or "WHEN trigger THEN action". A second type of applications stems from the outstanding work done with Yahoo's Pipes. These applications are based on the idea of providing a visual pipeline generator for supporting end users in creating aggregation, filtering, and porting of data originated by sources. The visual strategies adopted by following IFTTT or Yahoo's Pipes compliant solutions are promising techniques but, in our opinion, they present some lacks. The former offers a very simple and easy to learn solution based on the definition of ad hoc rules that can notify the end users when something happens – e.g. when their favorite sites are updated, when they check-in in some places or their friends do, or warn them when specific weather conditions are going to take place. However, the language is not enough expressive for the specification of more sophisticated rules based on time and space conditions. On the other hand, the latter offers a too complex solution for supporting the end user in expressing their preferences. Pretending that end users can deal with APIs of several sensors/devices put at risk the success of the visual approach. Moreover, events in each stream of an IoT scenario, are time and space dependent and so the related rules need to take into account these type of conditions. Nevertheless, in the described systems, time and space dimensions are almost neglected. Once the flow of events for a given analysis have been identified by IoT engineers, rules should be defined for specifying the action to be actuated when specific events occur. Our system, named Rule Editor, extends the IF-THIS-THEN-THAT approach and supports the definition of rules in a more articulated way by keeping the complexity at an acceptable level. The idea is to keep the simplicity of the IF-THIS-THEN-THAT system pairing it with the use of formula languages. Moreover, time and space dimensions are exploited and adopted for expressing more loose rules in the statements. From the point of view of language expressiveness, we consider rules that are more expressive than the IF-THIS-THEN-THAT system from different perspectives. Complex conditions

[4] http://www.drools.org/

[5] http://www.ibm.com/software/websphere/

are supported to correlate events both from the temporal and spatial dimensions. The time dimension allows end users to set rules that can be fired at some specific time, delayed in case of certain conditions are verified, and may be repeated until some event happens. The space dimension gives users the chance of linking rules to the place/area where the events currently taking place, where they will possibly be in the future or where they are moving into. Moreover, rules are enabled depending on spatio-temporal conditions that are clearly outlined in the rule specification. This allows the identification of the streams that are really relevant in a given time and given location for monitoring given situations and properly react to them. By exploiting a Rule Language, the rules can be specified by means of the graphical Rule Editor. As depicted in Figure 2 the graphical interface is based on drop-down menus that are populated by using the attributes that characterize the JSON of the flow of events produced by the IoT Engineers with the SmartLoader environment.

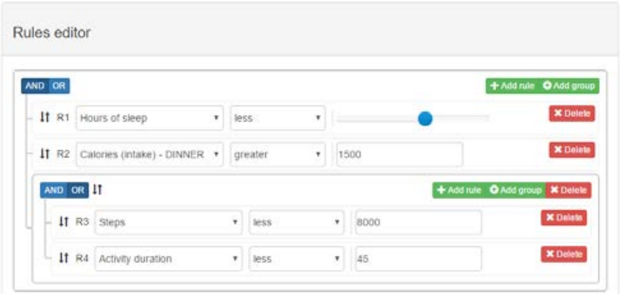

Figure 2. The composition of the rules of a CompositeRule.

Figure 3. The creation of a temporal condition in the Rule Editor system.

An example of Composite Rule creation is given in Figure 2, where the user has defined four rules: (i) R1: Hours of sleep less than 7 (ii) R2: Calories intake at dinner greater than 1,500 (iii) R3: Number of steps less than 8,000 (iv) R4: Activity duration less than 45 (minutes). The rules are built in this way: R1 AND R2 AND (R3 OR R4). The meaning of the Composite Rule created in this example is: "if the hours of sleep of the day before are less than 7 AND if the calories intake at dinner (before the sleep) is greater than 1,500 AND (if the number of steps is less than 8,000 at day OR the duration of physical activity is not less of 45 minutes at day THEN send the athlete a message that warns about the behavior and performances. The rule editor aims at allowing non-technical people to specify rules by using simple drop-down menus. The conditions can be composed by combining groups of statements connected by using the operator AND or OR. Temporal conditions are defined as depicted in Figure 3 and use the automatically assigned names of the Rules as elements to be composed (R1, R2, R3, ...). An example of temporal condition is that R3 (number of steps less than 8,000) has to take place before R2 (calories intake at dinner greater than 1,500) with a range of +/- 5 minutes. In other words, the dinner must begin within 5 minutes after the user has finished to walk (for less of 8000 steps). Another complex temporal condition can be: R4 (activity duration less than 45 minutes) begins before of the R2 (Calories intake at dinner greater than 1,500) and ends after +/- 10 minutes. In other words, the trainer wants to check if her/his athletes eat too much and how, if they eat not seated at a table but when they are on the move and quickly (in less than 45 minutes). Once a rule is created, it is stored in a repository for further re-use or for sharing it among members of a community of trainers and coaches. In this way, it easy for other users to customize existing rules according to their needs.

Conclusions and Future Works

In this paper, we have presented an approach for supporting IoT engineers, coaches and trainers in designing and analyzing data-flows coming from a set of physical/social sensors and services. At the moment we are working for setting up a user test evaluation for our eWellness system thanks to collaboration with the CSI (Centro Sportivo Italiano – Italian Sport Centre). In this use case, a set of sport teams will be involved in order to perform a set of tasks. What we want to study from the user tests, is how far our approach is able to offer new possibilities both at the design and use time and to understand how the idea to combine the design and end user environments appears to be successful and effective solution both for domain and technical experts.

References

1. Sadri, F. 2011. Ambient intelligence: A survey. ACM Computing Surveys 43, 4, 1-66
2. Lieberman, H., Paternò, F., Klann, M., and Wulf, V. 2006. End-User Development: An Emerging Paradigm. In End-User Development. Springer, 1-8.
3. Costabile, M. F., Mussio, P., Parasiliti Provenza, L. and Piccinno, A. 2008. End users as unwitting software developers. In Proc. of WEUSE 2008. ACM, 6-10
4. Costabile, M. F., Mussio, P., Parasiliti Provenza, L. and Piccinno, A. 2008. Advanced Visual Systems Supporting Unwitting EUD. In Proc. of AVI 2008. ACM, 313-316.
5. Barricelli, B.R., Marcante, A., Mussio, P., Parasiliti Provenza, L., Valtolina, S. and Fresta. G.: BANCO: a Web Architecture Supporting Unwitting End-User Development. IxD&A, 5-6, pp. 23--30 (2009).
6. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, AG. and Mehandjiev, N. 2004. Meta-Design: A Manifesto for End-User Development. CACM 47, 9, 33-37.
7. Wenger, E. 1999. Communities of practice: Learning, meaning, and identity. Cambridge university press.
8. Mesiti, M., Valtolina, S., Ferrari, L., Dao, M.-S. and Zettsu, K. 2015. An editable live ETL system for Ambient Intelligence environments. WF-IoT (2015), 393–394.
9. Valtolina, S., Barricelli, B.R. and Mesiti, M. 2015. End-User Centered Events Detection and Management in the Internet of Things. Current Trends in Web Engineering. Springer. 77–90.
10. Bauleo, E., Carnevale, S., Catarci, T., Kimani, S., Leva, M. and Mecella, M. 2014. Design, realization and user evaluation of the smartvortex visual query system for accessing data streams in industrial engineering applications, JVLC. 25, 5, 577-601.