A Visual Paradigm for Defining Task Automation

Carmelo Ardito
Università di Bari,
via Orabona, 4
70125 Bari, Italy
carmelo.ardito@uniba.it

Giuseppe Desolda
Università di Bari
via Orabona, 4
70125 Bari, Italy
giuseppe.desolda@uniba.it

Maristella Matera
DEI – Politecnico di Milanoo
Piazza Leonardo da Vinci, 32
20134 – Milano, Italy
matera@elet.polimi.it

Copyright is held by the author/owner(s).
AVI, June 07–10, 2016, Bari, Italy.

Abstract
In the last years, researchers are devoting many efforts to improve technological aspects of the Internet of Things (IoT), while little attention has dedicated to social and practical sides. Professional developers program the behavior of smart objects. In addition, often the functionality exposed by a single object are not able, alone, to exhaustively support the end users' tasks. The opportunities offered by IoT can be amplified if new high-level abstractions and interaction paradigms enable also non-technical users to compose the behavior of multiple objects. To fulfill this goal, we present a model to express rules for smart object composition, which includes new operators for defining rules coupling multiple events and conditions exposed by smart objects, and for defining temporal and spatial constraints on rule activation. Such model has been implemented in a Web application whose composition paradigm has been designed during an elicitation study with 25 participants.

Author Keywords
End-User Development of Mashups, Visual Paradigms for ECA Rule Expression, Internet of Things.

General Terms
Design, Human Factors.

Introduction
The current integrated technologies confer intelligence to any type of objects (also called *things* in this document) and connect them to the network. For example, a pressure sensor can make an office door handle smart, for example alerting a user located in another part of the world every

time someone comes into that office. The added value is that the object, being part of a network of *smart things* (the *Internet of Things* (IoT) can communicate with other objects or services, thus triggering dynamic behavior. If the door sensor is able to communicate with a sensor worn by a person, than a remote user can also know who entered the office.

One of the opportunities of the IoT is especially the possibility to collect in real time information concerning events and behaviors happening in the real world. For example, a potential advantage of the IoT is to anticipate the needs of a human even before he is aware of it.

Today these scenarios may appear unrealistic since we are not pervaded with a significant number of sensors. However, Cisco's forecasting estimates that more than 50 billion smart objects will be deployed by 2020[1]. There are already many contexts in which the IoT is adopted. For example, for creating *wearable devices*, i.e. clothing and accessories such as bracelets, watches, t-shirts, rings, shoes, can integrate sensors capable of detecting physiological parameters and actuators to communicate with the users. *Domotics* is another very active IoT context: pressure, volumetric and distance sensors, as well as infrared cameras, are able to ensure the security of homes; they permit a remote user to interact with the devices installed in the house (e.g., lights, heating, blinds) and assist user to avoid hazards such as floods, fires or explosions. Other sectors that are beneficiating of the advent of IoT are Smart Cities, industry and environment through energy saving.

[1] http://newsroom.cisco.com/feature-content?type=webcontent&articleId=1208342

IoT Challenges

From the technological point of view, such a large number of things requires an adequate network infrastructure and efficient communication protocols, especially due to the fact that integrated devices have very limited resources (e.g., CPU, RAM, memory, battery). Further issues, such as privacy (e.g., in the smart door scenario, if and when it is allowed to know who entered into the office) and safety (e.g., if a hacker is able to break a device, then it could access all the devices it is connected with) must be addressed.

From the HCI point of view, a major challenge is to enable even non-technical users to manipulate data and functionality of things in a simple and natural way. Today, in fact, this is a prerogative almost always reserved to developers who, through the use of specific programming languages, provide pre-packaged solutions to users. The most important challenge is to allow non-technical users to define and manage the connections between things, which represent the real benefit of IoT, especially in the next years when a growing amount of things will be available.

Some works in the literature propose mashup techniques for addressing this issue. For example in [1] and [3] the authors introduce two systems for the mashup of things for home automation, both consisting of two design environments: one is devoted to electrical engineers who define the behavior of devices through a visual representation of logic operations and algebraic formulas; the second one allows non-technical users to create a web page where they can include widgets to display data coming from things and synchronize their behavior based on a "wired" composition paradigm. The problem is however that several studies have demonstrated that this kind of composition paradigm is not suitable for non-expert users [5, 6, 8] as it forces them to deal with concepts like data flow and parameter passing which cannot be mastered by people who are not expert in programming. One of the products available in the market is *gluethings* (http://www.gluethings.com), a Web platform for registering and composing things. Unfortunately, also this system offers a wire-based composition paradigm and also require users to handle JSON to set parameters for the low-level behavior of things – such practices are out of reach for laypeople, i.e., those users which represent the actual business opportunity.

To determine the success of the IoT, it is necessary to investigate new approaches that, thanks to high-level abstractions, can enable non-expert users to compose data and functionality of things, as well as the communication among them, by means of "natural" composition paradigm.

Outlook on the composition paradigm

As mentioned above, things can be treated as services, because sensors and actuators have an URI that identify them on the Internet. In the case of service mashups, platforms implementing an event-driven approach, as the one described in [4], permit to synchronize Web services so that the event produced by/on a service (e.g., selection of a word in a text) triggers an action of another service (e.g., a search using the selected word). We believe that an event-driven paradigm is suitable also for the manipulation of things offering the advantage of enabling the composition of things among themselves and also with other Web services. Similar event-driven platforms (e.g., IFTT - https://ifttt.com/) are now emerging also for *task automation*. They make it easy to connect two services (things or APIs), choose a trigger, and thus create an action. They look very promising thanks to the simple and effective composition paradigm, even if it is intended for expert users and therefore requires programming skills.

In fact, most of these platforms currently used in the IoT domain do not permit (for example IFTTT) or make it difficult (for example *gluethings*) to specify multiple events and actions. Similar difficulties arise when specifying temporal or spatial constraints, e.g., to define the following behavior: "If *I'm in Rome* and I post an image on Instagram *between 8.00 and 11.00 a.m.*, post the same picture also on Twitter and Tumblr".

In order to identify composition paradigms able to guide users in the definition of articulated rules, we wanted to elicit the end-user mental model, which is an aspect scarcely explored in the field of task automation, as pointed out by [7]. As the seed of our investigation, we were inspired by the *5W* model, which is adopted in several domains, such as journalism and customer analysis, and more in general in problem solving, to analyze the complete story about a fact. It suggests describing a fact by answering the following questions:

- Who did it?
- What happened?
- When did it take place?
- Where did it take place?
- Why did it happen?

We adopted the 5W model in an elicitation study with 25 participants aimed at identifying, with the help of users, a notation for the specification of task automation rules. The customization of the 5W model, which we called *Rule_5W*, helped us to highlight the elements that are essential for creating complete meaningful rules for smart object composition. In the Rule_5W model, "Who" is replaced by *"Which"* for specifying the services involved in a rule. *"What"* indicates the triggered events, as well as the actions to be

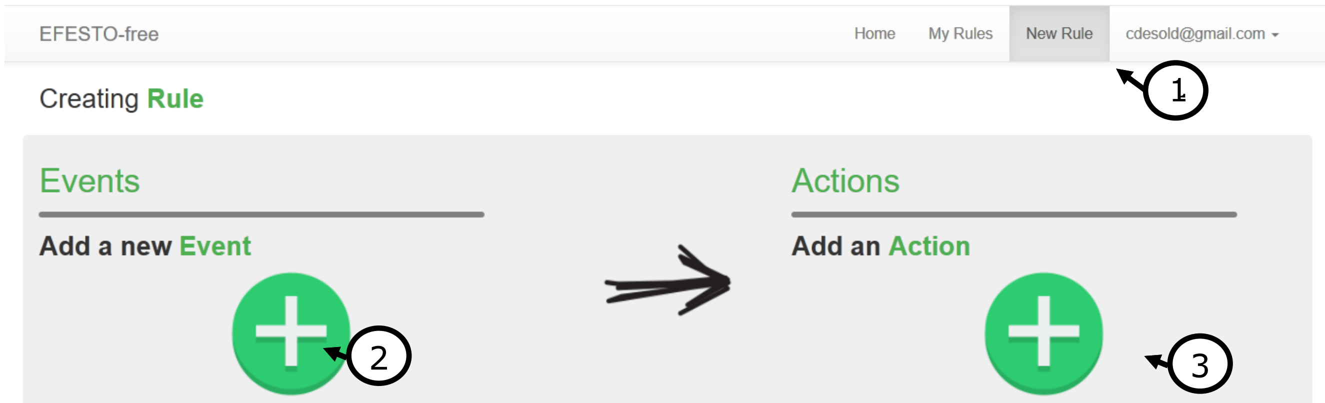

Figure 1. EFESTO: the interface for rule creation.

activated. *"When"* and *"Where"* refer to the specification of, respectively, temporal and spatial conditions for triggering events and performing actions. Finally, *"Why"* is used for reporting a short description to explain the rule behavior to a human reader, e.g., other users with whom the rule is possibly shared. The paradigm has then been implemented in a Web platform that extends the capability of EFESTO, a platform for the End-User Development (EUD) of Web mashups through which data provided by Web APIs can be integrated into unified visualizations [2].

EFESTO and its 5Ws Composition Paradigm

In this section we illustrate through an example the main features of the 5Ws composition paradigm. A user, who we suppose is a female, creates a rule to automatically turn on the coffee machine and roll-up the shutters when her smart bracelet detects that she has just woken up or the smart alarm clock rings. To create this rule, the user clicks the "New Rule" button in the navigation bar (**Errore. L'origine riferimento non è stata trovata.**, circle 1) and the "Creating Rule" interface appears. The UI shows the main area in which a rule is defined. The left side is for specifying the triggering events, and the right side is to define the actions to be activated by the selected services.

A **wizard procedure**, activated by the green "+" button highlighted by circle 2 in **Errore. L'origine riferimento non è stata trovata.**, guides the users in defining the events in a **full-automated** fashion. The wizard sequentially shows some pop-up windows in which the service, the events and the conditions are specified. According to **WYSIWYG** approach, the wizard steps allow the user to define an event in terms of *Which* is the service to be monitored for detecting the triggering event, *What* service event has to be monitored, *When* and *Where* the event has to occur. The specification of When and Where conditions is optional. At the end of the wizard procedure, the event is defined and its summary appears under the "Events" area.

Actions can be defined by clicking on the green "+" button highlighted (circle 3 in **Errore. L'origine riferimento non è stata trovata.**). The button activates a wizard that helps the user define an action in terms of *Which*

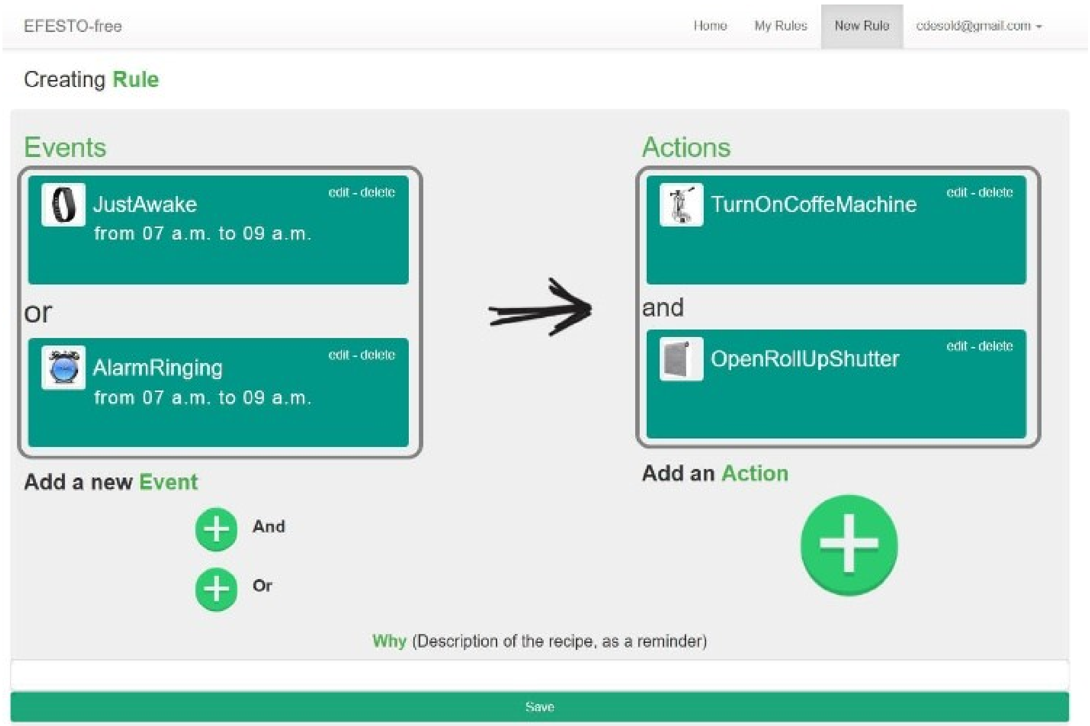

Figure 2. EFESTO: example of rule including two events and two actions.

service will execute the action as a consequence of the event(s), *What* action the service has to perform and *When* and *Where* the action can be performed.

In EFESTO, users may either define first all the events and then the actions, or define first a basic rule with one event and one action and then include new events and new actions. Events and actions can be added or removed at any time fostering a **dynamic modification of the running mashup**. Further events can be added by clicking one of the two green "+" buttons labeled And / Or. Choosing the "And" button starts the definition of a new event that will cause the execution of the rule action(s) if all conditions of all events are satisfied. The "Or" button determines the definition of a new event that will cause the execution of the rule action(s), if the conditions of at least one event are satisfied. Once the rule is created (see Figure 2), it can be saved by entering a short description of the rule (the *Why* in the 5W model).

Conclusion

One of the cornerstones of the future of the IoT will be to put in the hands of the end users simple software tools capable of making natural and powerful composition between things. It is unimaginable that this possibility is reserved for a few experts. As already happened in the past with software tools (forums, social networks, CMS) that made users to evolve from simple consumer to prosumer, even for the IoT have to be designed tools suitable even for non-expert users. This paper has illustrated how a generic platform for service mashups can be specialized for the composition of services that enable accessing/controlling smart things. We are currently working on implementing the needed extensions. A demo, illustrating some preliminary results, will be given during the workshop.

References

[1] Blackstock, M. and Lea, R. IoT mashups with the WoTKit. *Proc. IOT '12*. IEEE (2012), 159-166.

[2] Desolda, G., Ardito, C. and Matera, M. EFESTO: A platform for the End-User Development of Interactive Workspaces for Data Exploration. In *Rapid Mashup Development Tools - Rapid Mashup Challenge in ICWE 2015*. 591. Springer Verlag, 63 - 81, 2015.

[3] Guinard, D., Trifa, V., Mattern, F. and Wilde, E. From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices. In *Architecting the Internet of Things*. 1. Springer Berlin Heidelberg, 97-129, 2011.

[4] Matera, M., Picozzi, M., Pini, M. and Tonazzo, M. PEUDOM: A Mashup Platform for the End User Development of Common Information Spaces. In *Web Engineering - ICWE '13*. LNCS 7977. Springer, 494-497, 2013.

[5] Namoun, A., Nestler, T. and De Angeli, A. Conceptual and Usability Issues in the Composable Web of Software Services. In *Current Trends in Web Engineering - ICWE '10*. LNCS 6385. Springer, 396-407, 2010.

[6] Payne, S. J. Users' Mental Models: The Very Ideas. In *HCI Models, Theories, and Frameworks - Toward a Multidisciplinary Science*. Morgan Kaufmann, 135-156, 2003.

[7] Ur, B., McManus, E., Ho, M. P. Y. and Littman, M. L. Practical trigger-action programming in the smart home. *Proc. CHI '14*. ACM (2014), 803-812.

[8] Wajid, U., Namoun, A. and Mehandjiev, N. Alternative Representations for End User Composition of Service-Based Systems. In *End-User Development - ISEUD '11*. LNCS 6654. Springer, 53-66, 2011.