Giuseppe Desolda, Carmelo Ardito, Maristella Matera, Antonio Piccinno (2015):
Mashing-up smart things: a meta-design approach.
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. 55-60)

Mashing-up smart things: a meta-design approach

Giuseppe Desolda[1], Carmelo Ardito[1], Maristella Matera[2], Antonio Piccinno[1]

[1]University of Bari, Italy, [2]DEI –Politecnico di Milanoo, Italy
[1]*{giuseppe.desolda / carmelo.ardito / antonio.piccinno}@uniba.it*
[2]*matera@elet.polimi.it*

Abstract. Recent technology advances support the interconnection of smart things, enabling their communication according to the Internet of Things (IoT) paradigm. This paper discusses how the opportunities offered by such technologies can be amplified, by offering to the end users the possibility to define the behavior of their smart objects according to paradigms that capitalize on well-established approaches for mashup composition.

1 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, it is estimated that within 10 years the number of sensors will vary between 20 and 50 billion. 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, whichintegrate sensors capable of detecting physiological parameters. 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.

2 IoT Challenges

From the technological point of view, such a large number of things requires an adequate network infrastructure and efficient communication protocols, especially because 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.

Some works in the literature propose mashup techniques for addressing this issue. For example In [2] 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 sychronize their behavior based on a “pipes-like” composition paradigm. The problem is however that several studies have demonstrated that this kind of composition paradigm is not suitable for non-expert users [5-7] 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 few products available in the market is gluethings (http://www.gluethings.com), a Web platform for registering and composing things. Unfortunately, also this system offer a pipes-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.

3 A meta-design approach

Our position is that a meta-design approach can be adopted for the mashup of smart things in a similar way as we already proposed in [1] for the mashup of heterogeneous data sources. Smart things in the end produce data (for example parameters revealing the occurrence of given events). Considering the objects as sources of data is therefore reasonable. Moreover, composition approach for service mashups is based on an event-driven, publish-subscribe paradigms that suits very well the domain of smart objects whose behavior has to be synchronized.

We identify the configuration of things as a critical task, at least unless standard technologies and communication protocols are established. In this respect meta-design offer some advantages. Meta-design means designing for other designers and thus refers to the opportunity of creating software environments and tools that can allow other stakeholders (less experiences with programming) to becomedesigners themselves.

The meta-design approach we propose is characterized by three different layers, related to three distinct design phases offered to professional developers, domain experts and end users (see Figure 1). At the top layer, developers design and develop the design environments that enable the design by the end users.

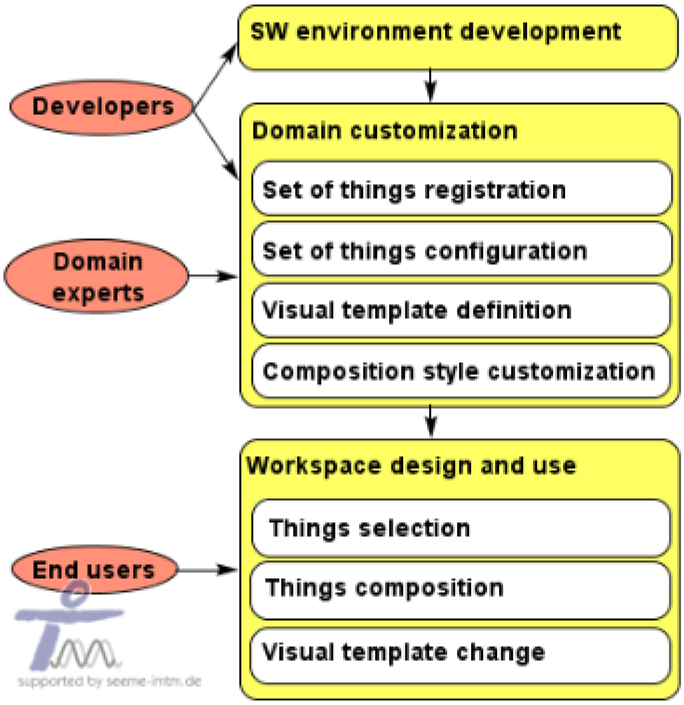

Figure 1: The three-layer model of our meta-design approach

At the middle layer, domain experts with the help of developers customize the platform to a specific domain. In the IoT context this implies performing the following activities:

- *Set of things registration:* sensors and actuators of a specific domain are registered. For example, in the case of home automation cameras, volumetricsensors, anti-burglary sensors, actuators to control the electric shutters, actuators to open and close the gas, etc;
- *Set of things configuration:*specification of which data and functionality are relevant for a specific domain. This activity may concern data sampling frequency definition (the data from the same sensor may be sampled in different ways depending on the context of use) or the definition of interfaces (a thing could be equipped with several sensors and actuators, and different users might have access only to a subset of them; for example a webcam installed in the center of a city can provide citizens with data streaming, while the administrator can also turn it on and off).
- *Visual template definition:* specification of the "visual containers" to be adopted by the end users to display and manipulate data extracted from things (such as charts, lists, maps). The provision of visual templates is a characterizing feature of our approach, as we propose composition paradigms where the customized visual templates provide the abstraction with respect to

technical details of service composition practice, which is also adequate with respect to the end users background and skills.

- *Composition style customization:* according to domain elements, such as users' skills and typical tasks performed with things, the overall composition style is also customized. Also, subsets of available compositions are identified and provided, in order to avoid the overloading the design environment. At the bottom layer, the final users are provided with an environment that permits not only to include and interact with predefined compositions of things (as in [2, 3]), but also to create new compositions. The user is able to define select data and functions exposed by sensors/actuators, and define how things have communicate among them.

4 Outlook on the composition paradigm

As mentioned above, things can be treated as services, because sensorsand 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 offeringthe 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 nowemerging 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. Our platform will also take advantage of such a simplicity deriving from the event-driven approach, which we already implemented for synchronizing APIs at the UI level. Additionally, we will take advantage of the meta-design methodology and of the intrinsic customizability of the platform. Our ultimate goal is to define a natural composition paradigms fitting the background and skills of the end users.

5 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.

6 References

[1] Ardito, C., Costabile, M. F., Desolda, G., Lanzilotti, R., Matera, M., Piccinno, A. and Picozzi, M. User-Driven Visual Composition of Service-Based Interactive Spaces. Journal of Visual Languages & Computing, 25, 4 (2014), 278-296.

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

[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] 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.