

# Empowering CH Experts to Produce IoT-enhanced Visits

C. Ardito, P. Buono, G. Desolda  
Dipartimento di Informatica,  
Università degli Studi di Bari Aldo Moro,  
Via Orabona, 4 – 70125 – Bari  
Italy

{carmelo.ardito, paolo.buono, giuseppe.desolda}@uniba.it

M. Matera  
Politecnico di Milano, DEIB  
Piazza Leonardo da Vinci, 32 – Milano  
Italy  
maristella.matera@polimi.it

## ABSTRACT

This demo presents EFESTO-5W, a platform for the definition of IoT-enhanced visits to Cultural-Heritage (CH) sites. Its main characteristic is an End-User Development paradigm applied to the IoT technologies and customized for the CH domain, which allows different stakeholders to configure the behavior of smart objects for creating more engaging visit experiences.

## CCS CONCEPTS

• Information systems~Mashups • Software and its engineering~Integrated and visual development environments

## KEYWORDS

End-User Development, Smart-Object Programming, Smart-Visit Experiences.

## 1 INTRODUCTION

The Internet of Things (IoT) has recently emerged as a technology able to enhance the access to CH collections. It promotes the use of *smart objects*, i.e., physical devices connected to the Internet, equipped with sensors and/or actuators and embedded software. Visitors of CH sites can touch, manipulate and interact with smart objects, for example to receive personalized information during the visit. Such interaction fosters the appropriation of CH content [7]: as recognized by some recent works [9], physical manipulation is indeed an effective supplementary channel for the visitor to gather and store information, since they are enabled to activate real-world knowledge.

In the CH domain, research has especially focused on technical aspects related to the development of sensor and actuator infrastructures [7]. A few works investigated how

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

UMAP'17 Adjunct, July 09-12, 2017, Bratislava, Slovakia

© 2017 Copyright is held by the owner/author(s).

ACM ISBN 978-1-4503-5067-9/17/07.

<http://dx.doi.org/10.1145/3099023.3099089>

non-technical users can configure smart objects to define smart visit experiences. To fill this gap, this demo proposes an End-User Development approach that we implemented within the EFESTO-5W platform [5], which empowers professional guides to enhance visit experiences by exploiting multiple synchronized smart devices.

End-User Development (EUD) methodologies fit very well the requirement of letting end users, with or without expertise in computer programming, to tailor, customize and program their software [2, 4, 6] or ecologies of smart objects [9]. Our approach proposes a visual composition paradigm that facilitates the creation of Event-Condition-Action (ECA) rules required to synchronize the behavior of multiple objects [5]. Its peculiarity is the expressiveness of ECA rules, which can be defined not only on native events/actions of smart objects, but also on properties that the designers of the smart experience add to the smart objects to assign them custom semantics. This additional semantics favors the creation of digital narratives that can be put in context with respect to the CH site content.

In the following, we describe the main characteristics of EFESTO-5W and how its architecture supports the development of different composition paradigms fitting to different domains. Afterwards, we briefly describe a demo showing a customization of the platform aimed to empower CH guides to define IoT-enhanced visits.

## 2 END-USER DEVELOPMENT OF SMART CH VISITS

EFESTO-5W is a Task-Automation System [3] that supports users without skills in computer programming to configure the behavior of smart objects by creating ECA rules [5]. The peculiarity of EFESTO-5W is the adoption of a model, called *5W*, that introduces some specification constructs (*Which, What, When, Where, Why*) to build ECA rules coupling multiple events and conditions exposed by smart objects, and to define temporal and spatial constraints on rule activation and actions execution. To foster the adoption by different communities of users, the architecture of EFESTO-5W features a decoupling of the interaction layer from the other platform modules. In particular, the aim

is to support the customization of the composition paradigm, by means of specific interaction metaphors that “speak the language of the user”, i.e., propose terminology, concepts, rules, and conventions the user is comfortable with.

As shown in Figure 1, the platform is organized in three layers, each one managing a separate aspect. The *Interaction Layer* refers to the system client; it handles the User Interface (UI) through which the users can create ECA rules. In particular, the *Rule Generator* is an interpreter that translates the user visual actions for rule creation into a rule descriptor file. The *Service Builder* is invoked each time an event/action, which is exposed by a service managing a smart object, has to be included in the rule. It is also in charge of representing in the UI the registered services available in the *Service Descriptor* repository.

At the server side, the *Logic Layer* manages rules and services by means of respectively the *Rule Engine* and the *Service Engine* modules. The first one receives the rule descriptor file from the client and instantiates a rule object based on a publish-subscribe, event-action model. The second one manages the requests on the available services coming from the Interaction Layer. It is also invoked by the Rule Engine when the rule is created, to verify if the rule events are satisfied and, in case, to activate the rule actions.

The *Service Layer* is located at the server side and stores service and rule descriptors by using JSON files. A service descriptor contains all the information useful to remotely query a smart object and contributes to decouple the registered services from the rest of the platform. A rule descriptor stores all the details of the instantiated rules and it is used when a rule needs to be instantiated.

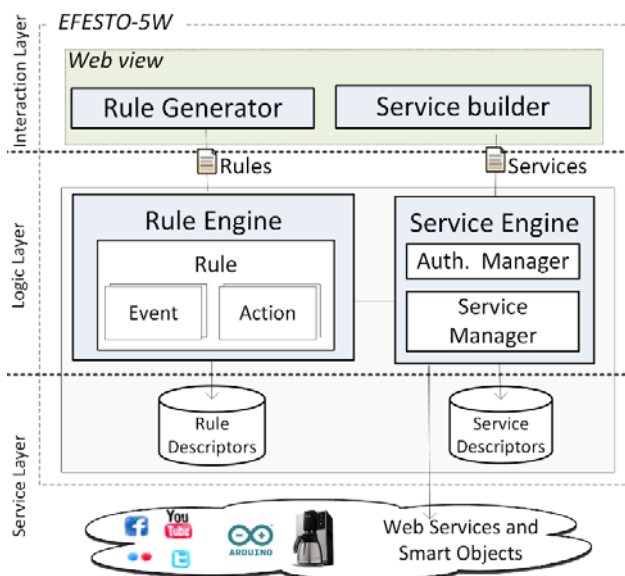


Figure 1. Overall organization of EFESTO-5W architecture.

### 3 DEMO DESCRIPTION

The demo will show how a CH curator can configure the behavior of an ecology of smart objects by using a customization of the EFESTO-5W platform for the CH domain. This customization was possible thanks to the separation of the UI layer from the other architectural layers. The configuration language is visual. Through it, CH experts can enhance the smart objects by means of sensible properties and define, on this basis, ECA rules that synchronize smart objects behavior with respect to physical and logical characteristics of the CH site captured during the visit. CH experts can thus create digital narratives that, by means of the sensing capability of smart objects, provide visitors with personalized content.

The demo will show the system at work on a set of smart objects that we recently designed for an archaeological park in Apulia (Italy): a smart version of the “Mouth of Truth”, and a deck of smart cards depicting archaeological finds. The demo will focus on the configuration of a smart visit and show how, by using the EFESTO-5W design environment, a CH expert can *i)* add into the system some custom properties describing the semantics of the smart objects and *ii)* define ECA rules governing the behavior of smart objects and the provision, within an interactive game, of content adapted to the different, evolving situations of use.

### REFERENCES

- [1] Ardito C., Buono P., Costabile M. F., Lanzilotti R., and Piccinno A. (2012). End users as co-designers of their own tools and products. *Journal of Visual Languages & Computing* 23(2), 78-90.
- [2] Coronado M. and Iglesias C. A. (2016). Task Automation Services: Automation for the Masses. *IEEE Internet Computing* 20(1), 52-8.
- [3] Costabile M. F., Fogli D., Mussio P., and Piccinno A. (2007). Visual Interactive Systems for End-User Development: A Model-Based Design Methodology. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 37(6), 1029-46.
- [4] Desolda G., Ardito C., and Matera M. (2017). Empowering end users to customize their smart environments: model, composition paradigms and domain-specific tools. *ACM Trans. on Computer-Human Interaction (TOCHI)* 24(2), Article 12 (April 2017), 52 pages.
- [5] Lieberman H., Paternò F., and Wulf V. 2006. *End User Development*. Springer.
- [6] Mighali V., Fiore G. D., Patrono L., Mainetti L., Alletto S., Serra G., and Cucchiara R. (2015). Innovative IoT-aware Services for a Smart Museum. In *Proc. WWW '15*. Florence, Italy, 18 - 22 May. ACM, 547-50.
- [7] Risseuw M., Cavada D., Not E., Zancanaro M., Marshall M., Petrelli D., and Kubitz T. (2016). An authoring environment for smart objects in museums: the meSch approach. In *Proc. SERVE 2016 in AVI 2016*. CEUR-WS, 25-30.
- [8] Uckelmann D., Harrison M., and Michahelles F. (2011). An Architectural Approach Towards the Future Internet of Things. In *Architecting the Internet of Things*, Uckelmann D., Harrison M., and Michahelles F. (Eds.) Springer Berlin, 1-24
- [9] Yannier et al. 2016 Yannier, N., Hudson, S.E., Wiese, E.S. and Koedinger, K.R. (2016). Adding Physical Objects to an Interactive Game Improves Learning and Enjoyment: Evidence from EarthShake. *ACM Trans. Comput.-Hum. Interact.* 23(4), 1-31.