Personal and Ubiquitous Computing
https://doi.org/10.1007/s00779-020-01457-5

ORIGINAL ARTICLE

User-defined semantics for the design of IoT systems enabling smart interactive experiences

Carmelo Ardito[1] · Giuseppe Desolda[2] · Rosa Lanzilotti[2] · Alessio Malizia[3,4] · Maristella Matera[5] · Paolo Buono[2] · Antonio Piccinno[2]

Received: 16 December 2019 / Accepted: 5 September 2020

Abstract
Automation in computing systems has always been considered a valuable solution to unburden the user. Internet of Things (IoT) technology best suits automation in different domains, such as home automation, retail, industry, and transportation, to name but a few. While these domains are strongly characterized by implicit user interaction, more recently, automation has been adopted also for the provision of interactive and immersive experiences that actively involve the users. IoT technology thus becomes the key for Smart Interactive Experiences (SIEs), i.e., immersive automated experiences created by orchestrating different devices to enable smart environments to fluidly react to the final users' behavior. There are domains, e.g., cultural heritage, where these systems and the SIEs can support and provide several benefits. However, experts of such domains, while intrigued by the opportunity to induce SIEs, are facing tough challenges in their everyday work activities when they are required to automate and orchestrate IoT devices without the necessary coding skills. This paper presents a design approach that tries to overcome these difficulties thanks to the adoption of ontologies for defining Event-Condition-Action rules. More specifically, the approach enables domain experts to identify and specify properties of IoT devices through a *user-defined semantics* that, being closer to the domain experts' background, facilitates them in automating the IoT devices behavior. We also present a study comparing three different interaction paradigms conceived to support the specification of user-defined semantics through a "transparent" use of ontologies. Based on the results of this study, we work out some lessons learned on how the proposed paradigms help domain experts express their semantics, which in turn facilitates the creation of interactive applications enabling SIEs.

Keywords Internet of Things · Smart Interactive Experiences · End-User Development of IoT interactive systems · Trigger-action programming · Ontologies · User study

1 Introduction

Automated systems have their roots in the field of automation, which has been defined by the Britannica encyclopedia as *the application of machines to tasks once performed by human beings or, increasingly, to tasks that would otherwise be impossible.*[1] Around the 1940s, the term automation was coined in the automobile industry to indicate the use of automatic devices in the mechanized production line. More recently, the expression task automation has been introduced to characterize systems that, in different domains, from leisure to work, take over the execution of tasks from humans to reduce the amount of work for a given task [20, 35].

In the last years, Internet of Things (IoT) has emerged as one of the most investigated technologies for task automation.

[1] https://www.britannica.com/technology/automation

✉ Maristella Matera
maristella.matera@polimi.it

[1] Department of Electrical and Information Engineering, Politecnico di Bari, Via Orabona, 4, 70125 Bari, Italy

[2] Computer Science Department, University of Bari Aldo Moro, Via Orabona, 4, 70125 Bari, Italy

[3] School of Creative Arts, University of Hertfordshire, Hatfield, Hertfordshire, UK

[4] Molde University College, Molde, Norway

[5] Department of Electronics, Information and Bioengineering, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20134 Milan, Italy

Published online: 26 September 2020

It is promoting the development of smart applications and smart environments in different domains and application areas, from home automation [23], to smart factories [49], to ambient assisted living [30], whose peculiarity is the orchestration of rules governing the automatic behavior of multiple devices. Some of such systems are fully automated; i.e., the orchestration of IoT devices and Web services is used to monitor the environment and automate some tasks consequently. The interaction by human agents is not necessarily needed, as these systems generally monitor environment variables (e.g., temperature, luminosity, user presence/proximity) to enact actions that can change the state of the physical environment, and more specifically of some physical devices installed in it (e.g., switching on the heating, opening the window blinds, switching on the lights). The user interaction is implicit [41]; for example, the user provides input by entering some rooms. Explicit user actions are still possible, and can be required to control the activation of some devices, for example, vocal commands for conversational agents controlling the home-automation system (e.g., Alexa), but are not essential for the functioning of the system.

Besides this class of IoT systems, which largely covers many needs related to everyday automation, novel systems are also emerging in the field of human-automation interaction [35], to promote immersive, sometimes multi-sensory, experiences where users are empowered to interact with the surrounding environment, also through smart objects and tangible interactions [33]. In several cases, content organized in narratives is also provided. In these interactive- and content-intensive systems, the user plays the role of a further agent whose behavior needs to be considered in the orchestration governing the system [31]. This technological landscape is the key for enabling *Smart Interactive Experiences* (SIEs—pronounced "see-ehs"), i.e., immersive experiences created by orchestrating different devices so that the resulting smart environments are completely automated to fluidly react to the usage situations induced by the situational needs of the final users [6].

The number of domains in which SIEs are provided for different purposes is growing; they range from education and learning [22, 57], to rehabilitation and health therapies [28], to Cultural Heritage (CH) [4, 54, 55]. Despite this diffusion, critical issues have to be solved to increase the practical impact of such applications, as it is still unclear how lay-people (e.g., non-technical domain experts such as educators, caregivers and therapists, museum curators) can be supported in their everyday work activities, being enabled "to harness the potential power of these large collections of devices to accomplish their tasks" [45]. Specific solutions support the automation of few smart objects [54], but they are not adequate for lay-people who want to automate a constellation of devices installed in the environment or held by final users.

By focusing on this gap, our research proposes an approach to support non-technical people to design *IoT-based automated systems* enabling SIEs. The approach that we present in this article has been extensively validated with domain experts in the CH who, in the context of their everyday work activities, are required to configure IoT systems supporting smart interactive visits of museums and CH sites. The main idea is to enable domain experts (in the following also called *designers*, because they create and modify systems enabling SIEs) to conceive and program such systems by starting from defining a semantic enrichment, deriving from their domain knowledge, of the resources involved in the final applications [4]. The advantage of such *user-defined semantics* is that the initial phase of enrichment can facilitate the reflection needed to conceive "in the large" the final application. The introduced semantic terms can then facilitate the following "in-the-small" design, where domain experts are required to automate the behavior of IoT devices by defining Event-Condition-Action (ECA) rules. The literature recognizes the value of semantic enrichments, e.g., through ontologies, for facilitating the event-driven programming of IoT devices also in other domains. Some approaches propose to annotate smart devices with ontology concepts to improve the semantics of ECA rules [18]. However, the proposed paradigms have a limited expressiveness with respect to the background and expertise of non-technical people. In addition, there is a scarce or completely missing consideration and systematization of the user-defined semantics; this limits the possibilities for designers to adopt ontologies and exploit their benefits.

To alleviate these problems and foster the creation by non-technical people of applications for SIEs, we recently proposed an approach that promotes user-based semantics as a conceptual tool for the domain experts to conceive such applications creatively. In [4], we reported on the evaluation of a visual composition paradigm that allows designers to freely define properties aimed to assign semantics to smart objects. The results of this study, which focused especially on the usability of the visual environment for the definition of semantic properties, pushed us to investigate further paradigms for supporting and stimulating the designer's creativity. To this aim, we implemented three systems, two of which are based on tangible user interfaces (TUIs) while the third one implements Augmented Reality mechanisms. These systems have been evaluated during a wide study that allowed us to collect a broad spectrum of data, allowing us to investigate multiple research perspectives. For example, in [6], we analyzed the quantitative data purposely collected to identify trade-offs existing between six different quality dimensions (Creativity, Workload, UX, Engagement, Utility, Completeness, Ease of Use). The results of this analysis address the perspective of developers of tools for the design of interactive IoT systems, who need to choose the most adequate interactive technologies that best suites specific design contexts and also maximize the considered quality dimensions.

Concerning our previous results, this article provides two novel contributions. First, it describes how the architecture of an existing tool for ECA rule definition [23] has been enriched through the integration of ontologies, to support the management and systematization of user-defined semantics. This is achieved through an algorithm that transparently assists designers in using and extending a domain-specific ontology through their semantic properties. Second, this article analyzes the qualitative data collected during the study presented in [6], this time to investigate the perspective of non-technical designers on (1) the role of the ontologies in the definition of the user-defined semantics, (2) the support that each system provides to facilitate the definition of the user-based semantics and, more in general, the design of IoT systems enabling SIEs.

This article is organized as follows. The next section discusses the rationale and background of our work concerning the related literature. The third section presents a software architecture for semantically enriched systems to be used for designing for SIEs. It also illustrates an algorithm to assist the semantic enrichment of IoT devices through ontologies. The fourth section briefly describes the systems implementing the three interaction paradigms that creatively support the semantic enrichment. The study comparing the three systems is presented in the fifth section, while some lessons learned are discussed in the sixth section. The seventh section concludes the article by highlighting future work.

2 Rationale and background

In the last years, research on IoT-based automated systems has primarily focused on technical aspects. However, to foster the appropriation of this technology by people operating in different application domains, a very important issue is how to enable them to directly tailor and evolve smart environments, without any (or with limited) intermediations by software professionals [27]. This relates to the "End-User Development for the Internet of Things," a topic addressed by a recent special issue [45] that offers a picture of this fast-evolving research field. Indeed, some works are addressing the problem of enabling end-users to automate the behaviors of smart devices, through paradigms simplifying the definition of *Event-Condition-Action* (ECA) rules [2, 10, 12, 23, 30, 46].

In the market, the so-called Task-Automation (TA) tools also follow this direction [12, 20, 23, 68]. Two types of TA tools emerge: (a) tools that support users in the creation of "basic" rules, such as rules that synchronize one event with an action; IFTTT is the most popular tool in this category; (b) tools that permit users to specify complex rules through paradigms that are not adequate to non-technical end-users. Examples in this second category are Node-RED [37], Microsoft Flow [47], and Crosser [21].

Intending to simplify the definition of ECA rules, while allowing a reasonable level of complexity in the rules, some works propose the use of ontologies to build a semantic layer where high-level concepts provide an abstract and technology-independent representation of the smart devices [17, 18, 66]. In this way, ECA rules can be defined on top of the ontology classes leaving out the technical details of smart objects. For example, in IFTTT, automation can be expressed as "IF the position of my smart bracelet is Monega Road 23, London THEN on the Nest Thermostat set 22 °C." Thanks to ontologies, the same rule would be simplified as "IF I enter home THEN set the home temperature to 22 °C" [18]. This semantic enrichment, however, requires experts to create ontologies and to define mappings between ontological concepts and the involved smart devices. Our approach, instead, allows designers to express the semantic enrichment of smart devices through a transparent use of ontologies that does not require technical skills.

2.1 User-defined semantics

An ontology is an explicit specification of an abstract view of the world, which contains objects, concepts, and other entities that exist in some area of interest, and the relations among them [32]. Ontologies are useful in several fields to limit complexity and organize information into knowledge (e.g., digital libraries, and semantic web). However, their use in real contexts is still limited due to the skills and effort required for their management. To foster a wider adoption, different frameworks and tools assist ontology creation and exploration [38] by means of visualization techniques like *indented list* (e.g., Protégé [51], OntoEdit [65], and OntoRama [26]), *node-link* (e.g., OntoViz [64], SpaceTree [56], OntoTrack [43], GoSurfer [71]), and *3D structures* (e.g., *Cone Tree* [58], OntoSphere [8], Reconfigurable Disk Tree [36]). However, ontologies have been also adopted to annotate Web services to facilitate their integration in systems requiring third-party components. For example, the *METEOR-S* framework provides support to semi-automatically annotate Web services with concepts of domain-specific ontologies [53]. Automatic approaches have been proposed also to annotate RESTful Web services [61] and services in the cloud [59]. Since full-manual annotation would require a significant effort, these approaches certainly ease entity annotation; however, their precision is not accurate.

The research reported in this article aims to support designers of applications for SIEs in managing domain-specific ontologies capturing their knowledge and use them to annotate smart devices. This work builds on previous research that investigated the value of *user-defined semantics* as a means to simplify the design of IoT interactive systems. In [4], a visual framework was proposed to empower the designers to build a semantic layer for ECA-rule definition based on the specification of *custom attributes* (*CAs*), i.e., properties that allow domain experts to externalize their tacit knowledge [11]. CAs are meant to add knowledge that can simplify the definition of ECA rules [17]. Their peculiarity is that they are defined by the designers, who are the end-users in our design approach (in this sense, we refer to them as *user-defined* terms), and relate to specific usage situations. These two aspects help designers of applications for SIEs make sense of the available digital resources by contextualizing them in a specific usage situation.

To better understand the notion of CAs and their contribution to the design of applications for SIEs, in the following section, we present a scenario where the SIE to be supported is an interactive game designed by two professional museum guides.

2.2 Motivating scenario

Mary and Tony are professional guides of an archeological park hosting an exhibition on the tools adopted by ancient populations that lived in that area in the past. During their tours, their activity mainly consists of showing and explaining the usage of the exposed tools to groups of pupils visiting the park. To improve the exhibition, Mary and Tony create a game that, through the use of IoT devices, automates the explanation of the park contents during the visit and, through an interactive game-play, aims to make the visit more engaging and effective. In a dedicated area of the park, pupils get a collection of tools (tangible interactive objects that are replicas of significant ancient tools), which are digitally augmented through RFID tags. Then, during the game-play, they are asked to identify tools having a specific characteristic, for example, those used in a *specific activity* of the populations (e.g., cooking, fighting, building). They have to position the tools on one of the 3D models installed in the area to reproduce some aspects of the ancient activities (e.g., eating/praying moments, typical places for such activities), and augmented with an RFID reader. The 3D models are characterized by other peculiar characteristics. Therefore, the game proposes group quests, such as “Put on the *Messapian kiln* all the *fighting tools* of the *same age*.” When pupils move the right tool on the target 3D model, this plays an audio file illustrating the tool features, and the pupils get points as a reward. Otherwise, the 3D model plays an audio file indicating that the selected tool is wrong. The game continues with pupils answering all the proposed quests. The winner is the group that gets the highest score.

Before automating the behavior of smart devices (e.g., the 3D models and the smart tools) through ECA rules, according to the approach presented in this article, for each interactive object, Mary and Tony define properties (attributes of the object) that express the object meaning and role according to the game dynamics. For example, each tool has descriptive attributes, like “Age” (with values: *iron*, *bronze*, *Messapian*, *Roman*), “Activity” (with values: *cooking, fighting* and *building*), “Material” (with values: *gold*, *pottery*, and *bronze*), and “Audio file” (with values indicating names of soundtracks to be played in case of a right answer). Mary and Tony “freely” define these attributes and their values according to the purpose of the game: for each attribute they want to create, the system asks them to type an attribute name, and then it shows a list of attributes, extracted from domain-specific ontologies that are semantically related to the typed name (see the next section for more details on the recommendation feature). The users can finally choose one attribute among the ones proposed by the system or they can keep the starting one. This is why we call these terms *custom attributes*.

After specifying CAs, Mary and Tony create the ECA rules to automate the behavior of the smart devices. To this aim, they can use a visual paradigm, like the one proposed in EFESTO-5W [23], to create ECA rules. An example of ECA rule based on CAs, represented here in a textual syntax, is:

$$Rule_i\text{: “IF } userTool.\text{isOn}(3DModel) \text{ AND}$$
$$userTool.\text{Age} = 3DModel.\text{Age THEN}$$
$$3DModel.\text{playsAudioFile}(userTool.audio\_file)\text{”.}$$

Assigning attributes to smart objects has two main advantages when creating ECA rules. First, the language adopted to define the rules is closer to the domain-expert language. For example, the variables occurring in the rules are the attributes created by Mary and Tony. Second, the attributes introduce abstractions that favor generalization. Without CAs, several rules would be defined for every single smart device, like:

“IF tool_012.isOn(3DModel_5) THEN
3DModel_5.playsAudioFile(explanation04.mp3)”.

This rule would be replicated for each tool and each 3D model, and the designer has to remember the ID of any tool and 3D model. Supposing that the same SIE involves *n* tools and *m* 3D models, the first rule covers all the *nXm* possible automation rules. Several systems for IoT programming instead force users to create all the combinations [20].

3 Back-end for user-defined semantics

This section illustrates the architecture of the EFESTO-4SIE platform (see Fig. 1); in particular, it describes the back-end mechanisms for handling CAs on top of ontologies. The

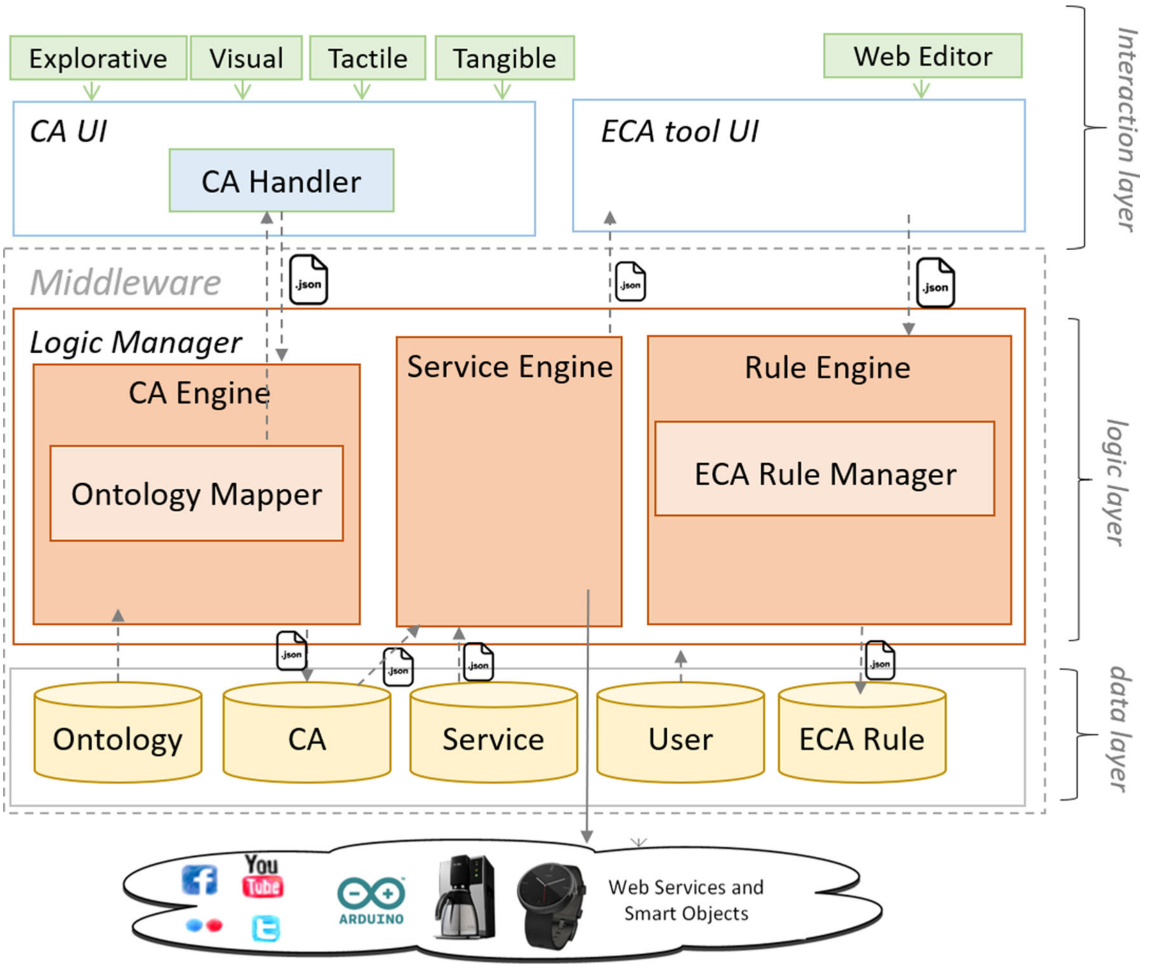


Fig. 1 Architecture of the EFESTO platform

supported design activities are two: (1) the definition of CAs and (2) the automation of smart objects through the creation of ECA rules. The modules supporting the second activities are described by Desolda et al. [23]. In this section, we focus on the management of the user-defined semantics, which is the novel aspect discussed in this article. It is worth noting that the peculiarity of the integrated platform is the possibility to "plug-and-play" different front-ends implementing different interaction paradigms, all exploiting the same back-end. This feature facilitated our study on the different interaction paradigms for CA definition that we discuss in the following sections.

In order to favor the decoupling between the platform back-end and different front-ends, the *Interaction Layer* requires the implementation of a *CA Handler* module for the definition of CAs. This module listens to the actions performed by the users when defining CAs, and converts them into *CA descriptors* saved in a *CA repository* available in the data layer. CA descriptors store information like *type* (e.g., textual or numerical), *name* and *value*, the *user* who has created the CAs, and the *smart devices* a CA is associated with.

The system front-ends invoke RESTful API functions available in a middleware that implements the *Logic* and *Data Layer*. The first layer consists of a *Logic Manager* that implements three modules, each one exposing different APIs. The *CA engine* is in charge of validating CA descriptors (e.g., checking if the user ID and the smart device ID exist respectively in the User Descriptor and Service Descriptor repositories). Through the *Ontology Mapper*, it also enables linking the user-defined CAs with classes of a domain-specific ontology. In our current implementation, we adopted the CIDOC CRM ontology [1] that provides definitions and a formal structure for describing concepts and relationships used in Cultural Heritage documentation [1]. A domain-independent vocabulary is also used to consider general-purpose terms [48].

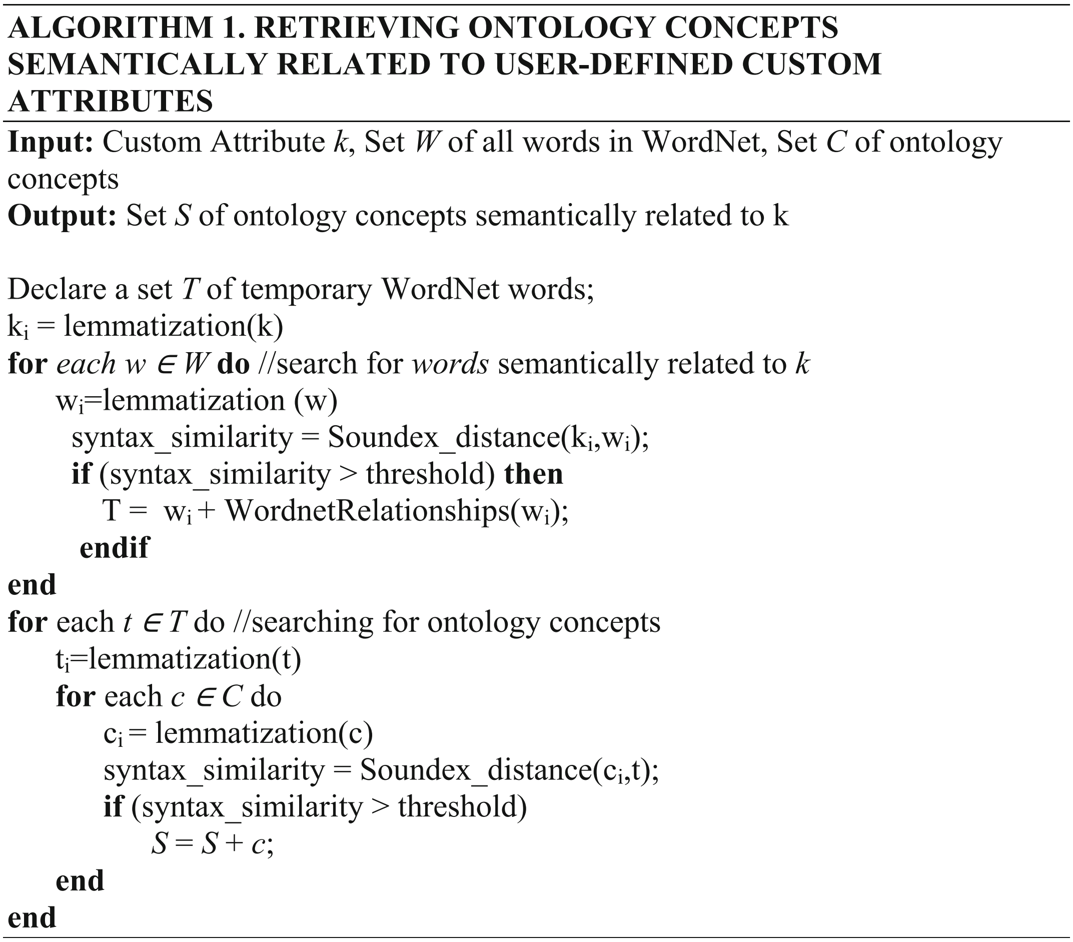
Thus, the Ontology Mapper (1) checks if the defined CAs are semantically and syntactically correct according to the reference domain, (2) offers suggestions for additional or alternative suitable CAs that users had not thought of, and (3) provides an organization of the defined terms that can allow domain experts to share CAs with others interested in defining other applications for similar SIEs.

3.1 Suggesting custom attribute names

As soon as the designer enters a CA into the system, the platform provides suggestions for refinements based on attributes retrieved in the domain-specific ontologies. This is achieved through the computations summarized in Algorithm 1. The goal is to retrieve a set of ontology concepts whose semantic is similar to CA *k* initially expressed by the designers so that they can choose the term that best describes the CA meaning.

The algorithm initially identifies a set of words available in a lexical ontology that are semantically similar to the keyword(s) typed by the designer. In our current implementation, the lexical ontology is WordNet and the semantic relationships we are interested in are synonymy, hypernym, hyponym, holonymy, and meronymy [48]. The other semantic relationships provided by Wordnet are not used in our current implementation; indeed, an empirical evaluation of our algorithm showed that, if including them, most of the concepts retrieved by the algorithm result to be not relevant. Lemmatization is then calculated for both *k* and each *w* in *W* before their comparison. To this aim, we adopted the Stanford CoreNLP library, which provides a set of human language technology tools, for example, to retrieve the base forms of a word as for lemmatization. In order to compare the lemmatized forms of *k* and *w*, the Soundex distance is adopted [72]; if their similarity is more than 0.7/1 (this threshold has been calculated during a tuning phase of the algorithm performed on around 100 trials), *w* is added into the set *T*.

After that, the algorithm finds all the ontology concepts whose lemmatized name is syntactically similar to the lemmatized words in *T*. The retrieved concept names are included in the set *S,* which is the output of the algorithm. This list of candidates CAs is sent back to the CA Handler module, which presents them to the user. The user can choose one of the suggested terms as CA name. Alternatively, the user can decide to keep the original name in two modalities: (1) associating it to one of the suggested terms or (2) as a new class in case none of the suggested terms is considered relevant. In the first case, a new *similar_to* association is added into the ontology, while in the second case, a new class is created and associated with the ontology root. This enables extending the ontology through the user contribution.



ALGORITHM 1. RETRIEVING ONTOLOGY CONCEPTS SEMANTICALLY RELATED TO USER-DEFINED CUSTOM ATTRIBUTES

Input: Custom Attribute *k*, Set *W* of all words in WordNet, Set *C* of ontology concepts
Output: Set *S* of ontology concepts semantically related to k

```
Declare a set $T$ of temporary WordNet words;
$k_i$ = lemmatization(k)
for each $w \in W$ do //search for words semantically related to $k$
    $w_i$=lemmatization (w)
     syntax_similarity = Soundex_distance($k_i$,$w_i$);
     if (syntax_similarity > threshold) then
         T =  $w_i$ + WordnetRelationships($w_i$);
      endif
end
for each $t \in T$ do //searching for ontology concepts
    $t_i$=lemmatization(t)
    for each $c \in C$ do
        $c_i$ = lemmatization(c)
        syntax_similarity = Soundex_distance($c_i$,t);
        if (syntax_similarity > threshold)
            $S = S + c$;
    end
end
```

Table 1 Summary of the methods that the Service Engine instantiates for each CA, according to its type. The first row reports methods valid for all the three attribute types, i.e., numerical, textual, and location. The other two rows relate to numerical and location attributes

	Events	Actions
All types	[CA_Name]_equals_to(p) [CA_Name]_different_from(p)	[CA_Name]_ assign_val(p) [CA_Name]_remove() [CA_Name]_reset()
Num	[CA_Name]_less_than(param) [CA_Name]_greater_than(param) [CA_Name]_in_range(p1, p2)	
Loc	[CA_Name]_inside(coord, radius) [CA_Name]_outside(coord, radius)	

3.2 Exploiting CAs in ECA rule definition

As shown in the motivating scenario, the *Interaction Layer* also supports the creation of ECA rules based on previously defined CAs. Every time the users select a service (e.g., a smart device), the *ECA UI* invokes the *Service Engine* in the Logic Manager, which returns a descriptor that specifies (i) all the events and actions natively provided by the service and (ii) further events and actions purposely built by the Service Engine by considering the CAs defined on a service. Table 1 reports the methods that can be generated, depending on the CA type. The ECA UI shows these events/actions, and the user can select them to specify the ECA rule, which is finally saved in a descriptor. The *Rule Engine* executes the rule by creating a rule object based on a publish-subscribe, event-driven model [13, 14, 23].

4 Front-ends for user-defined semantics

After showing how the integrated architecture helps handle (i.e., systematizing and structuring) the user-defined semantics, in this section, we illustrate three interaction paradigms conceived to facilitate the creative expression of the user-defined semantics by domain experts. A design workshop with 28 non-technical users (11 females, mean age 26 years), in groups of 5 or 6, was carried out to elicit interaction paradigms for SIE design [5]. In particular, the goal of the workshop was to elicit novel interaction paradigms that, besides assisting users in easily defining CAs, could stimulate designers' creativity, as emerged in [4]. To this aim, we guided the workshop groups to elicit three paradigms following some principles widely recognized in the literature as stimuli for creative design:

1) Tangible user interfaces (TUIs) augment the real physical world by coupling digital information with everyday physical objects and environments. Manipulation of physical artifacts improves tangible thinking, which is the ability to think through the manipulation of objects augmented with digital information [34].
2) Tactile interaction engages multiple users to interact with physical and virtual objects at the same time and privileges natural and intuitive social interactions. Most of the computer-mediated approaches for creativity support single-user interactions, thus failing to account for collaboration in group-based, face-to-face scenarios [15]. To overcome this problem, we considered tabletops and tangible objects interaction, which in literature are considered suitable for creative collaborative processes [15, 25, 39].
3) Exploratory search is effective in stimulating creativity [62], especially if the users can walk during the search activity [52].

No further methodological or technological constraints were imposed on the groups. However, we could not expect that they would come out with a complete and successful design proposal. Thus, we adopted a scenario-based design to maintain an orientation towards the workshop goal [60].

Each group was involved in three elicitation sessions, one for each paradigm. Each session started with a 10-min introduction given by an HCI researcher to introduce participants to the addressed domain. Possible examples of technological or methodological solutions were not shown to avoid any bias in the participants' proposals. During each session, participants were stimulated by two moderators (senior HCI researchers) to reason on two specific aspects of the paradigm, i.e., (1) how to represent Cas, and (2) how to associate CAs to smart objects. Participants were invited to follow a scenario similar to the one reported above in the second section [60]. Ideas proposed by participants were sketched and eventually complemented with textual descriptions.

At the end of the workshop, we collected five proposals for each paradigm, one for each group. The two workshop moderators analyzed the proposals. Video recordings of the workshops were also examined to avoid losing important details. Sketches having features in common were integrated into a unique design proposal, also by considering interaction

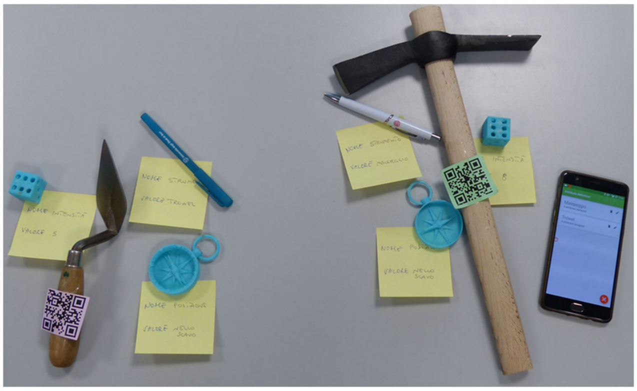


Fig. 2 The Tangible System: tangible attributes, post-it notes, and a smartphone are used to define the CAs

principles and usability criteria. At the end of this phase, three different proposals emerged and gave rise to the three prototypes described in the following sections.

4.1 The tangible system

The *Tangible System* proposes interaction with tangible objects. The rationale behind this paradigm is that it can stimulate creativity during design, thanks to the manipulation of physical objects augmented with digital information [34]. Two types of tangibles objects are used: (i) the *smart devices* to be used during the execution of the final application; (ii) other tangible objects that represent CAs; we refer to them as *tangible attributes*.

During the design workshop, the participants identified three main types of CAs, *textual*, *numerical*, and *locational*, and three corresponding tangible attributes for their "physical" representation: a *pen* for *textual* attributes, a *dice* for *numerical* attributes, and a *compass* for locational attributes. The idea that most largely emerged for CA definition consisted of exploiting the co-proximity of tangible attributes and smart devices plus specifying the attribute name and value by using post-it notes attached to the tangible attributes. For example, as shown in Fig. 2, close to the trowel (the main object on the left side), the designers put a pen to define a textual attribute, and a post-it on the pen to specify the name and value of the CA, e.g., "*Activity* = *build*." The designers perform the same actions for each CA. To store the tangible composition in a digital form, designers use a mobile app to take a picture of all the elements on the table. This app implements pattern recognition and deep learning techniques we developed to automatically recognize all the elements on the table (smart devices, tangible attributes, and post-it text) [7]. Such elements are then automatically converted by the app into the definition of <*CA name* = *value*> pairs. Once the CAs are in place, the designers can create the ECA rules, like the one of the motivating scenario, by using a visual interface such as the one proposed in [23].

4.2 The explorative system

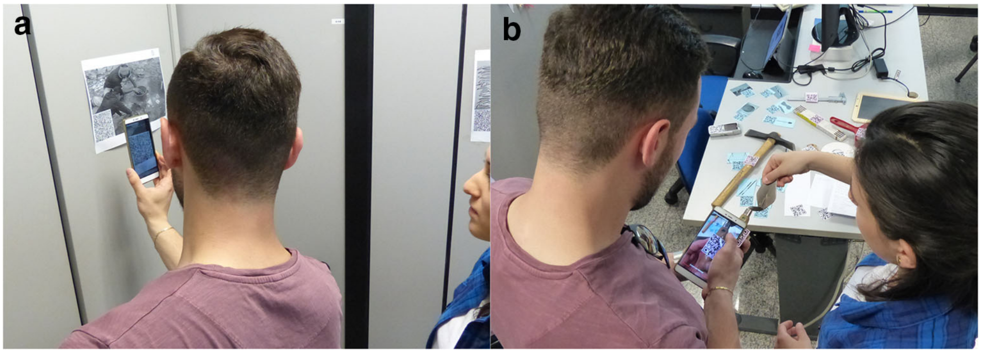
The *Explorative System* is based on the idea of the interactive exploration of surrounding environments, where *source objects* can suggest CAs enriching the smart devices of the final interactive application. Designers frame the source objects with the camera of the mobile device and see them in an AR fashion, with a virtual layering of properties that may suggest CAs (Fig. 3a). The presented properties can be "copied and pasted" onto the smart devices by using the same mobile app, this time framing the smart devices. From a pop-up menu listing the source object properties, designers can select one or more attributes that can be then associated with the smart devices by scanning its own QR code (Fig. 3b).

4.3 The Tactile System

The *Tactile System* combines tangible user interfaces with tactile interaction. A tabletop multi-touch is conceived as a digital workspace where designers perform the association of CAs with smart devices.

The same tangible attributes used in the Tangible System, i.e., the pen, the dice, and the compass, have been adopted to represent types of CAs. To associate smart devices with CAs, the users have to put a tangible attribute close to a smart device on the tabletop; this action will be detected by the system that reacts showing on the screen a text box where the users can

Fig. 3 In the Explorative System, a mobile app is used to explore the environment searching for attributes provided by source objects (**a**) and associate the attributes to smart devices to be used by the final interactive application (**b**)



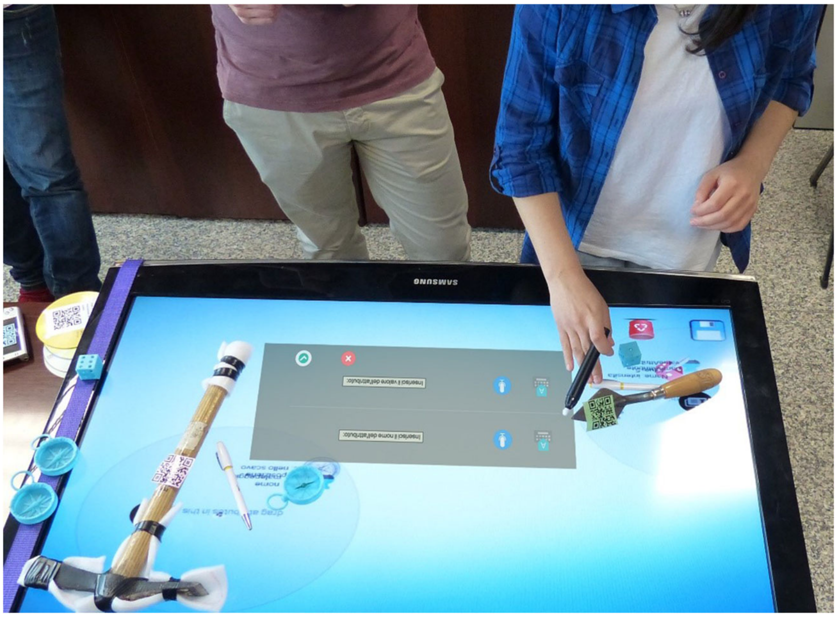


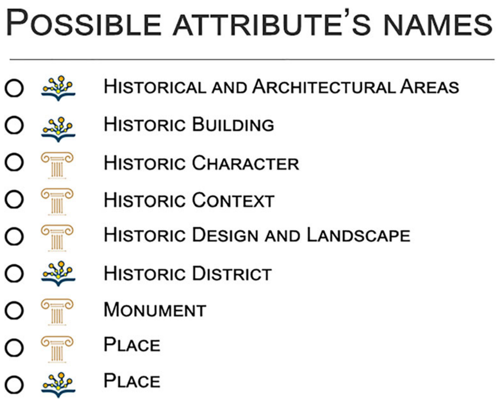
Fig. 4 The Tactile System: a tangible attribute like a pen is put close to a smart object on the tabletop; the system shows on the screen a text box where the users can specify the attribute name and value using a virtual keyboard or speech transcription

type the attribute name and value instead using post-it notes (Fig. 4).

5 Custom attribute refinement through ontologies

As detailed in the previous section, when the users define a CA, the Ontology Mapper module proposes a list of CAs semantically related to the entered term.

All the three systems visualize the resulting CA names in a dropdown list (see Fig. 5). The users can select the most adequate name for the CA under the definition, or they can decide not to use any of the suggested terms. In the second case, the system asks them to necessarily associate the new CA to one of the suggested terms: a new class is created in the ontology and the selected term is associated with it. If the users do not associate the new CA to any term, a new class is created and associated with the ontology root.



POSSIBLE ATTRIBUTE'S NAMES

○ HISTORICAL AND ARCHITECTURAL AREAS
○ HISTORIC BUILDING
○ HISTORIC CHARACTER
○ HISTORIC CONTEXT
○ HISTORIC DESIGN AND LANDSCAPE
○ HISTORIC DISTRICT
○ MONUMENT
○ PLACE
○ PLACE

Fig. 5 Example of ontology classes (part of the entire list) found by the algorithm and visualized by the three systems in reaction to the user keyword "Heritage"

6 User study

According to our approach, design for SIEs is a process characterized by two main phases, i.e., the creation of CAs and the definition of ECA rules. Since we already assessed the validity of the paradigm for ECA rule creation [23], this article focuses on the CA creation phase. To this aim, an experimental study has been carried out to understand how the interaction paradigms described in the previous section support non-technical users in the CA definition. To cover the entire process, and verify whether CA definition would support designers both in the initial conception phase and in the following design activities focusing on the dynamics of the target interactive application (i.e., the automation of the corresponding smart objects), participants were also required to write the ECA rules on a sheet reporting empty templates of rules. We choose Cultural Heritage (CH) as the reference domain since in the last years there has been an increasing interest by CH experts in installing smart objects into museums and CH sites to foster content fruition and visitors' engagement [54, 70].

Therefore, the research question driving this study is: *Do the three systems support designers to easily define CAs for the design of SIE-enabling applications?*

Through this study, we collected several quantitative and qualitative data that allowed us to analyze multiple research perspectives. In [6], we focused on the perspective of design tool developers. We identified and discussed some correlations that emerged from the quantitative data analysis, which led us to define trade-offs that developers of tools for SIE design can consider. In this article, we focus on the qualitative data collected to analyze the perspective of another stakeholder, i.e., the designers of applications for SIEs. In particular, we concentrate on (1) the role of the ontologies during the creation of CAs, and (2) the support each system provides in defining CAs and, more in general, in creating applications for SIEs. Such data have been analyzed by performing a thematic analysis that led out to draw lessons learned on the adoption of user-based semantics in design for SIEs. To clarify, we briefly report the study apparatus, which is deeply described in [6].

6.1 Participants and study design

The study involved a total of 18 participants (13 females, mean age 24 years) recruited from students attending a master's degree course in CH of the University of Bari in Italy. Participants were randomly organized into nine groups of two. As resulting from the demographic questionnaire they filled in during the study, 10 participants already attended at least one excavation campaign, spending a good amount of days in those kinds of professional activities; 5 of them also organized guided tours in museums or archeological parks; the remaining 3 participants did not have any type of CH professional activities. All participants have a good experience in IT and in using mobile devices, interactive displays, and smart devices. The independent variable "system" (3: *Tangible*, *Explorative*, *and Tactile*) was manipulated within-subjects.

6.2 Procedure

We installed the systems with their apparatus in three quite rooms at the Computer Science Department of the University of Bari. Each participant was immediately asked to sign a consent form that reported the study purpose. After the agreement, they filled in a demographic questionnaire. The study was organized along with three phases.

First phase: each group conceived an interactive IoT application following a scenario that asked them to act as museum curators who have to plan a new exhibition to disseminate the value of the archeological investigation. The interactive IoT application was required to be inspired to an interactive visit on the stratigraphic investigation exposing the involved tools and multimedia material. This first phase facilitated the ideation of the smart interactive visit by adopting brainstorming techniques [40, 67]. For each group, the entire phase lasted about 1 h.

Second phase: every group used the three systems (with the order counterbalanced according to a Latin Square design to avoid carry-over effect), one at a time, to design the smart interactive visit. Participants defined the attributes ideated during the first phase or introduced further attributes if needed, with the support of the recommendation features. Assessing the ECA rule creation was not the objective of this study [23], thus to simplify and speed up the process each participant wrote down the ECA rules on a paper sheet guided by rule templates. This phase lasted 15 min. Eventually, they filled in an online questionnaire about the system they used. Some open questions asked participants' feedback about the attribute recommendation since each system helped them during the attribute definition by suggesting additional names every time they defined a new attribute. After using the three systems, participants filled in a paper questionnaire to compare user satisfaction with all of the systems.

Third phase: each day of the study, once all the three groups completed the second phase, the participants attended together a focus group. The discussion revolved around a list of topics, like the experience in using the systems, and in particular, on the definition of custom attributes supported by the recommendation algorithm integrated into the system.

6.3 Data collection

A broad spectrum of quantitative and qualitative data was collected, whose analysis allowed us to investigate multiple perspectives. Quantitative data representing Creativity, Workload, UX, Engagement, Utility, Completeness, and Ease of Use, was previously analyzed to find trade-offs that can be useful for the development of design tools for applications for SIE [6].

This article focuses on the analysis of qualitative data, to investigate the perspective of the final applications on (1) the role of the ontologies during the creation of CAs, (2) the support that each system provides to the definition of the user-based semantics, and, more in general, to the design of applications for SIEs. We collected (1) transcriptions of audio recordings and notes from the second phase, (2) the notes taken by the observer on significant behaviors or externalized comments of the participants during the second phase, (3) the answers to the open questions included in the questionnaire, (4) transcriptions of the focus groups' audio recordings, and (5) the notes taken by the observer on significant externalized comments made by the participants during the focus groups. Two senior HCI researchers analyzed these data in a systematic qualitative interpretation using Inductive Thematic Analysis [9]. The two researchers independently double-checked the results. The initial reliability value was 75%; thus, the researchers discussed the differences and reached a full agreement. In the following section, we summarize the findings from our analysis.

6.4 Results of the thematic analysis

This section presents the themes developed through the analysis of the qualitative data collected in the study. For each theme, the most explanatory participants' utterances are reported.

Theme no. 1. Custom attribute recommendation. From the workshops, it emerged that all the participants agreed on the usefulness of recommendations helping CAs definition. For example, 9 participants highlighted that suggestions on CA alternatives foster a more creative and informed decision since "some terms may not be thought during the design but might be more meaningful than the conceived ones." Similarly, 7 participants said that "typos or grammar errors are resolved since the system suggests the right term and also useful alternatives."

Theme no. 2. Taking inspiration from the environments. The ability to define CAs starting from ideas caught in the environment (e.g., through the *Explorative System*) was considered a stimulating ingredient for creativity. For example, the comments from 6 participants (e.g., "We did not think to this attribute, it can be useful in our SIE," "Let's try to scan further pictures, maybe they can suggest us new ideas for new attributes") and the higher number of CAs defined with the *Explorative System* [6] highlighted that the exploration of adequate environments allows designers to collect useful ideas with a low effort, thus increasing the creativity.

Theme no. 3. Tangible attributes are a source of inspiration. Participants discussed a lot on the use of tangible attributes in the *Tangible* and *Tactile System*s. Five participants proposed additional tangible attributes, e.g., "a meter to specify numbers indicating measures, since the dice specifically evocate numbers used for game points." An important aspect that emerged from the discussion, as observed by six participants, is that the physical representation of the attributes is important to conceive new attributes. They reported, for example, that "tangible attributes stimulated the conception of attributes they did not have in mind during the initial brainstorming." One participant said that "the dice suggested to introduce points and to organize the visit as a serious game," while another participant said that "the compass suggested the introduction of locations." Four participants also suggested improving this aspect by "providing a larger set of tangible attributes."

Theme no. 4. Cross-device interaction enriches the design experience. Albeit participants were able to design SIE-enabling applications by using one system at a time, the focus groups revealed that the combination of the different proposed systems might be a key factor to further improve the design process. Indeed, almost all the participants highlighted that the systems have complementary peculiarities that can facilitate creativity in different ways. For example, to be even more creative, 6 study participants suggested to "freely switch across different systems" according to different factors, in a cross-device fashion.

Theme no. 5. Pros and cons of the Tactile System. Simplicity, interactivity, and collaboration were recurring keywords used by participants to underline how the Tactile System was simple and engaging, especially for collaborative tasks. The physical surface of the display was considered a "useful space to sort out ideas" by 3 participants. Also, the interactivity of the tabletop combined with the manipulation of tangible attributes creates overall a more pleasant interaction. A problem highlighted by 5 participants regards the size of the virtual keyboard, sometimes "too large and thus overlapping and hiding important parts of the user interface." Four participants also stressed two remarkably limiting factors for the adoption of this type of system in real contexts: its price, which is currently not so affordable, and the physical space that is required to install it.

Theme no. 6. Pros and cons of the Explorative System. This solution was appreciated because "it is very simple to use" and "it is a very cheap and compact solution since it requires only a smartphone." Three participants also said that "it simplifies the association of custom attributes to smart devices with a big size," like a "smart painting" that cannot be easily moved from the room where it is exposed. Three other participants affirmed that it can stimulate new ideas of attributes while they explore the environments. Given its explorative nature, it was also perceived as "very similar to the archaeologist work" by four participants. Two participants remarked that the small size of the smartphone screen does not provide a wide and clear overview of the smart devices and their custom attributes. Indeed, even though users can see the overview in a summary section, its fruition is not so immediate as in the other two systems. They suggested the adoption of a tablet instead to enlarge the display area for the augmented reality layer.

Theme no. 7. Pros and cons of the tangible system. Eight participants appreciated the small size of the *Tangible System* (a participant said that "it can be placed in a box after each use as a board game") and the reduced cost, which is an important factor in the CH domain. The use of post-it notes resulted intuitive for attribute creation, even if participants have to perform two actions (writing the attributes and taking the picture)—while in the *Tactile System*, they perform just one action. Six participants liked taking a picture for digitalizing the physical composition: a couple of participants said that "this feature is useful" since they can focus on the creation and association of the attributes in a "physical way," similar to what they did during the first phase of the study, thus more prone to the conception. Five participants also appreciated the ability to create compositions in the physical world ("We can use a desktop that is larger than the tabletop surface") and that they could associate multiple attributes at the same time and keep an overview of them. Regarding the limitations of this system, 4 participants highlighted that "the attribute creation with the Tactile System was slower as they had to write manually CA name and value on post-it notes."

7 Discussion and lessons learned

This experimental study helped us understand if and how our approach supports lay-people in designing automated systems for SIEs. This is a novel aspect, scarcely investigated in the field of human-automation interaction [35]. Indeed, existing approaches for task automation in IoT support the orchestration of few smart objects [20] but do not fit domains and application areas where lay-people want to automate complex and rich ecosystems of IoT devices installed in the environment or carried on by final users. This aspect goes in the same direction as the themes developed in this research field [35]. In particular, it relates to the "radical changes to human-automation interaction" that is needed to extend the design of automated IoT-based systems to non-experts; indeed, hitherto, it has been a prerogative of professional developers.

In the following, we discuss some lessons learned that, in line with the focus of the article, mainly refer to what we have observed for the definition of CAs. However, as discussed in the concluding section, the aspects highlighted in this work can drive researchers and developers to create systems supporting the design of interactive IoT applications by lay-people.

Introduce abstractions on smart objects. This first lesson learned mainly derives from observing that by using our approach all the 9 groups were able to define meaningful interactive applications for SIE. The development of tools supporting lay-people to easily automate a smart environment is a critical issue that poses different challenges. Caivano et al. recently performed a literature review on this class of tools, highlighting design implications for the development of future tools, like the need for *speaking the user's language* (e.g., the domain experts' jargons) and for *providing different levels of complexity* [12]. Such challenges are addressed in our approach by defining CAs and by using them in the ECA rule definition, for the designers to build a conceptual model of the SIE that is described by their terms. Regardless of the adopted interaction paradigm, we observed that defining an abstract and structured layer on top of the smart devices helps designers focus on and refine the SIE they want to support. It is worth mentioning that the use of CAs for ECA rule definition was not mandatory in the study; it was a feature that we provided to the participants and that they freely exploited in almost all cases. Indeed, CAs provide means for the domain experts to express and represent their tacit knowledge into the system [11], which in turn leads to an incremental building of domain-oriented repositories of concepts; those can be fed by domain experts themselves, reused across several design sessions, and shared between experts of the same domain to support collaborative design [42]. This is also in line with some recent works, which propose the adoption of ontologies to simplify the definition of ECA rules [18]. Defining CAs the designers are comfortable with, and overcoming at the same time the limitation of existing approaches for semantic enrichment of IoT systems [18], can be a flywheel for facilitating the design of IoT applications for SIEs.

Adopt ontologies transparently. As explained in the previous point, the adoption of CAs associated with ontology concepts is a way to facilitate the creation of ECA rules automating a constellation of smart devices. Given the vast amount of information typically available in ontologies, it becomes crucial to foster a transparent modality for their adoption; indeed, the use of ontologies does require technical skills that might lead to situations where the designers might not be able to cover their actual needs. As reported in theme no. 1 in the previous section, the recommendation approach proposed in this article resulted effective and was appreciated by the participants. Indeed, although the ontologies used in the study contain more than 1000 terms/classes, we observed how the algorithm simplifies their selection since designers have to deal only with a small set of concepts semantically related to the terms they initially define. This is also in line with the results of other studies related to the adoption of ontologies for recommending terms or concepts (see for example [24]) aimed to facilitate the definition of ECA rules [16, 19].

Stimulate creativity. One of the most important aspects of the approach presented in this article is the support offered to domain experts to creatively conceive IoT-based automated applications. The design of our systems for CA definition has been driven by principles that typically foster creativity in design processes [63]. The effectiveness of our design choices has been primarily confirmed by measuring the system support for creativity through the CSI (Creativity Support Index), a psychometric questionnaire that revealed that all the three systems strongly support designers when creating CAs [6]. More details on these scores emerged by the thematic analysis. For example, theme no. 2 and no. 6 highlighted the appreciation of participants for the Explorative System, since *source objects* installed in the environment stimulated further ideas of CAs. Installing the design apparatus in an inspiring environment is fundamental, as it represents a source of hints. The *Explorative System*, and in general the exploration paradigm, could be successful in environments like archeological parks that, by themselves, provide a rich source of attributes. This is in line with some findings on the design of creativity-support tools reported in [63]. Similarly, theme no. 3, no. 5, and no. 7 confirmed a wide appreciation of tangible interaction as a means to creatively define CAs. This is in line with works in literature that reports on the advantages of manipulating physical objects augmented with digital information [34].

Finally, while the previous points echo the literature, a new insight of our study is in theme no. 4: a combination of different solutions stimulating creativity in different ways, like the systems reported in this article, can be a key factor to boost overall creativity in the design process. For example, if designers can start the collection of attributes within a stimulating environment (e.g., at excavation sites), the *Explorative System* can be used at the beginning of the process; then the *Tactile* can allow designers to refine the initial ideas. The design approach proposed in this article, and in particular the enabling software architecture, makes it easy the implementation of the cross-device behavior since natively the interaction paradigms share the same back-end where all the elements of the final applications (CAs and ECA rules) are stored.

The workspace overview matters. Another aspect that influenced the creative design of CAs is the workspace overview. Modeling applications for SIEs can be difficult in the absence of a clear overview of the smart devices and the CAs being created. The overview can indeed support the verification phase of a creative process, which involves making sure that one's creative insight is an appropriate solution to one's problem [63, 69]. The *Tangible System* and *Tactile System* use a large physical space (a tabletop and a desk) where the different "tangible" elements can be positioned, and provide the overview without the need of switching to different modalities, as in the case of the *Explorative System*. The observation of the participant interactions with the three systems offering different workspace overviews leads us to underline the importance of considering proper overview mechanisms to favor continuity in the thinking flow (see themes no. 5, no. 6, and no. 7). For example, a wider overview, like the one provided by the *Tangible System*, can improve the designer's satisfaction with the creative process. Indeed, the physical space of the desk allows designers to compose spatially a larger number of smart devices and see them all at the same time, thus obtaining a complete, "tangible" overview on the design status. This increased control of the application under design makes designers more satisfied with their creativity because they can see (and also manipulate) what they are creating. In the experimental analysis, this was demonstrated by the correlation existing between the CSI index and the number of smart devices designed with the *Tangible System* [6].

8 Conclusions

The novelty in this work is the promotion of *user-defined semantics* to simplify the definition of IoT systems supporting SIEs. This contribution is motivated by the recurrent need, observable in different domains and especially in Cultural Heritage, to automate some tasks that involve user interaction and the provision of content within smart environments.

The definition of the user-based semantics introduces advantages for domain experts dealing with this class of IoT systems since it helps them contextualize the automatic behavior of smart devices within the narrative they want to convey to the final users of the interactive applications. This peculiarity distinguishes interactive IoT systems from fully automated IoT systems where user interaction and content provision are not relevant concerns. The focus on user interaction has been scarcely investigated in the field of End-User Development of IoT systems [23], and is the main original aspect of our work.

The article also discussed the results of a study that compared three interactive paradigms conceived to support the identification and specification of user-based semantics. These results confirm the validity of the proposed systems with our research question, and they led us to identify some interesting implications for improving design tools. In line with the focus of this article, the discussed implications mainly refer to what we have observed for the definition of CAs. However, it is also possible to identify aspects that, more in general, can help researchers and developers create tools for the design of interactive IoT applications by lay-people:

- One emerging aspect refers to the adoption of abstractions to represent the automation capabilities of IoT devices, as a means to conceive and modeling the final IoT applications. This vision suggests that it is important to extend the design of smart objects beyond hardware and software and identify high-level concepts that, by abstracting from the underlying technologies, can empower non-programmers to make sense of IoT devices and automate their behavior within applications adopted in their reference domains. Adequate conceptual models can thus serve as a knowledge base representing the building blocks of the End-User Development of interactive IoT systems [23].
- In relation to the need for abstractions, ontologies can be exploited to suggest concepts that can facilitate the definition of ECA rules for task automation. As also highlighted by other studies [16, 24], for lay-people to take advantage of ontologies, it is important to provide recommendation mechanisms that can help the exploration of the vast amount of available concepts. These concepts can be used also to recommend ECA rules. This is an aspect that has been already investigated [19] and that will be the goal of future extensions of the approach described in this paper.

- The interaction paradigms through which designers can master the different concepts and configure IoT devices also play a fundamental role. Our study showed that some interaction paradigms can support, more than others, specific phases of the creative design of IoT interactive applications. For example, the Tangible System has been considered more useful to support the SIE conception than the other two paradigms (see theme no. 7); furthermore, the combination of paradigms can be even more effective. Improving the support to creative design is fundamental to inspire people who are not used to designing (see theme no. 3, no. 5, and no. 7). In this respect, the choice of design material plays a central role. Our future research will investigate whether the adoption of a card-based co-design toolkit (e.g., [3, 29, 50]) can further improve this aspect.
- Design tools should also be organized so that designers can be in control of what they define. This also refers to the need of providing "debugging" mechanisms that can help designers understand the effect of their device configurations and test whether the modified or created behavior of the final applications renders as expected [44]. This is important also to prevent security and privacy issues that can be generated by the execution of user-defined ECA rules, especially in those domains where incorrect behavior of applications or actuators might have safety-critical consequences (e.g., in ambient-assisted living, or in-home automation).

Understanding how the previous aspects apply to domains different than Cultural Heritage will be the object of our future research: field studies, for example in the ambient assisted living domain, would be useful to further assess the generalizability of the results reported in this article.

Our future work will also focus on some limits that we identified in this study. From a technical perspective, we plan to improve the mechanisms for suggesting and enriching custom attributes. For example, the automatic recognition of objects in the surrounding environment could serve the retrieval of related content, by similarity matches, from online repositories [23].

Regarding the generalizability of the results, although we purposely configured the study environment to be as realistic as possible, we recognize the need to perform studies in the wild, in real design settings. This would allow us to further validate the impact that the environment exploration might have on creativity.

Finally, further studies should also address the output of the design process, i.e., the final applications for SIEs. The experience of the end-users, namely the museum visitors in the scenario presented in this article, should be evaluated to understand which factors influence the quality of the applications and how the design process can be refined to maximize it.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement. This work is partially supported by the Italian Ministry of University and Research (MIUR) under grant PRIN 2017 "EMPATHY: EMpowering People in deAling with internet of THings ecosYstems." We also acknowledge the financial support of MIUR through the PON project TALIsMAn - Tecnologie di Assistenza personALizzata per il Miglioramento della quAlità della vitA (Grant No. ARS01_ 01116).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Cidoc CRM. Retrieved from http://www.cidoc-crm.org/ Last Access June 10, 2019
2. Akiki PA, Bandara AK, Yu Y (2017) Visual simple transformations: empowering end-users to wire Internet of Things objects. ACM Transaction on Computer-Human Interaction 24(2):1–43
3. Ambe AH, Brereton M, Soro A, Chai MZ, Buys L, Roe P (2019) Older people inventing their personal internet of things with the IoT un-kit experience. In Proc. of the Conference on Human Factors in Computing Systems (CHI '19). ACM, paper 322
4. Ardito C, Buono P, Desolda G, Matera M (2017) From smart objects to smart experiences: an end-user development approach. International Journal of Human-Computer Studies 114:51–68
5. Ardito C, Desolda G, Lanzilotti R, Malizia A, Matera M (2017) Advanced interaction paradigms to define smart visit experiences in the Internet of Things era in Proc. of the Conference on the Italian SIGCHI Chapter (CHItaly '17), CEUR Workshop Proceedings, Vol-1910, 1148-1152
6. Ardito C, Desolda G, Lanzilotti R, Malizia A, Matera M (2019) Analysing trade-offs in frameworks for the design of smart environments. Behav Inform Technol 39(1):47–71
7. Balducci F, Buono P, Desolda G, Impedovo D, Piccinno A (2020) Improving smart interactive experiences in cultural heritage through pattern recognition techniques. Pattern Recogn Lett 131: 142–149
8. Bosca A, Bonino D, Pellegrino P (2005) OntoSphere: more than a 3D ontology visualization tool. In Proc. of the Italian Semantic Web Workshop (SWAP '05). CEUR workshop proceedings
9. Braun V, Clarke V (2006) Using thematic analysis in psychology. Qual Res Psychol 3(2):77–101
10. Brich J, Walch M, Rietzler M, Weber M, Schaub F (2017) Exploring end user programming needs in home automation. *ACM Transaction on Computer-Human Interaction* 24(2) 11 (2017):1–35

11. Budweg S, Draxler S, Lohmann S, Rashid A, Stevens G (2010) Open design spaces: socially crafting interactive experiences. In *Proc. of the ACM Conference on Designing Interactive Systems (DIS '10)*. ACM, New York, pp 442–443
12. Caivano D, Fogli D, Lanzilotti R, Piccinno A, Cassano F (2018) Supporting end users to control their smart home: design implications from a literature review and an empirical investigation. J Syst Softw 144:295–313
13. Cappiello C, Matera M, Picozzi M (2015) A UI-centric approach for the end-user development of multidevice mashups. ACM Transaction Web 9(3):1–40
14. Cappiello C, Matera M, Picozzi M, Sprega G, Barbagallo D, Francalanci C (2011) DashMash: a mashup environment for end user development. In: Auer S, Díaz O, Papadopoulos G (eds) *Web Engineering - ICWE 2011.* Lecture Notes in Computer Science, vol 6757. Springer Berlin, Heidelberg, pp 152–166
15. Catala A, Jaen J, Dijk Bv, Jordà S (2012) Exploring tabletops as an effective tool to foster creativity traits. In Proc. of the International Conference on Tangible, Embedded and Embodied Interaction (TEI '12). ACM, New York, NY, USA, 143–150
16. Corno F, De Russis L, Monge Roffarello A (2019) A high-level semantic approach to end-user development in the Internet of Things. International Journal of Human-Computer Studies 125: 41–54
17. Corno F, De Russis L, Roffarello AM (2017) A high-level approach towards end user development in the IoT. In Proc. of the SIGCHI Conference on Human Factors in Computing Systems - Extended Abstracts (CHI '17). ACM, New York, pp 1546–1552
18. Corno F, Russis LD, Roffarello AM (2017) A semantic web approach to simplifying trigger-action programming in the IoT. Computer 50(11):18–24
19. Corno F, Russis LD, Roffarello AM (2019) RecRules: recommending IF-THEN rules for end-user development. ACM Trans. Intell. Syst. Technol., Article 58
20. Coronado M, Iglesias CA (2016) Task automation services: automation for the masses. IEEE Internet Comput 20(1):52–58
21. Crosser. *Crosse*r. Retrieved from https://crosser.io/platform/crosser-node/. Accessed 28 Feb 2020
22. Delprino F, Piva C, Tommasi G, Gelsomini M, Izzo N, Matera M (2018). ABBOT: a smart toy motivating children to become outdoor explorers. In *Proc. of the Conference on Advanced Visual Interfaces (AVI '18)*. ACM, New York, pp 1–9
23. Desolda G, Ardito C, Matera M (2017) Empowering end users to customize their smart environments: model, composition paradigms and domain-specific tools. ACM Transactions on Computer-Human Interaction 24(2), Article 12 (April 2017), 52 pages
24. Desolda G, Matera M, Lanzilotti R (2020) Metamorphic data sources: a user-centric paradigm to consume linked data in interactive workspaces. Futur Gener Comput Syst 102:992–1015
25. Doering T, Beckhaus S, Schmidt A (2009) Towards a sensible integration of paper-based tangible user interfaces into creative work processes. In Proc. of the SIGCHI Conference on Human Factors in Computing Systems - Extended Abstracts (CHI '09). ACM, New York, pp 4627–4632
26. Eklund P, Roberts NP, Green S (2002) Ontorama: browsing an rdf ontology using a hyperbolic-like browser. In Proc. of the First International Symposium on CyberWorlds (CW '02), 405–411
27. Fröhlich P, Baldauf M, Meneweger T, Erickson I, Tscheligi M, Gable T, Ruyter BD, Paternò F (2019) Everyday automation experience: non-expert users encountering ubiquitous automated systems. In Proc. of the SIGCHI Conference on Human Factors in Computing Systems - Extended Abstracts (CHI '19). ACM, New York, pp 1–8
28. Garzotto F, Beccaluva E, Giannotti M, Riccardi F (2020) Interactive multisensory environments for primary school children. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '20)*. in print
29. Gennari R, Matera M, Melonio A, Roumelioti E (2019) A board-game for co-designing smart nature environments in workshops with children. In Proc. of the International Symposium on End User Development (IS-EUD '19). Springer international publishing, New York, pp 132–148
30. Ghiani G, Manca M, Paternò F, Santoro C (2017) Personalization of context-dependent applications through trigger-action rules. ACM Transaction on Computer-Human Interaction, 24(2), Article 14 (April 2017), 33 pages
31. Gilman E, Davidyuk O, Su X, Riekki J (2013) Towards interactive smart spaces. Journal of Ambient Intelligence and Smart Environments (JAISE) 5(1):5–22
32. Gruber TR (1993) A translation approach to portable ontology specifications. Knowledge Acquisition - Special issue: Current issues in knowledge 5(2):199–220
33. Guo B, Zhang D, Wang Z, Yu Z, Zhou X (2013) Opportunistic IoT: exploring the harmonious interaction between human and the Internet of Things. J Netw Comput Appl 36(6):1531–1539
34. Ishii H, Ullmer B (1997) Tangible bits: towards seamless interfaces between people, bits and atoms. In Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '97). ACM, New York, pp 234–241
35. Janssen CP, Donker SF, Brumby DP, Kun AL (2019) History and future of human-automation interaction. International Journal of Human-Computer Studies 131:99–107
36. Jeong C.-S, Pang A (1998) Reconfigurable disc trees for visualizing large hierarchical information space. In Proc. of the IEEE Symposium on Information Visualization (INFOVIS '98). IEEE computer society, Washington, DC, pp 19–25
37. JS Foundation. *Node-RED*. Retrieved from http://nodered.org/ Last Access February 28th, 2020
38. Katifori A, Halatsis C, Lepouras G, Vassilakis C, Giannopoulou E (2007) Ontology visualization methods—a survey. ACM Comput Surv 39(4):10
39. Kim MJ, Maher ML (2008) The impact of tangible user interfaces on spatial cognition during collaborative design. Des Stud 29(3): 222–253
40. Lamm H, Trommsdorff G (1973) Group versus individual performance on tasks requiring ideational proficiency (brainstorming): a review. European Journal of Social Psychology (EJSP) 3(4):361–388
41. Leifer L (2008) The design of implicit interactions: making interactive systems less obnoxious. Des Issues 24:72–84
42. Li Z, Barbara RB, Claudia I (2013) A meta-design model for creative distributed collaborative design. In: Nik B (ed) Development of distributed systems from design to application and maintenance. IGI Global, Hershey, pp 252–268
43. Liebig T, Noppens O (2004) Ontotrack: combining browsing and editing with reasoning and explaining for owl lite ontologies. In Proc. of the International Semantic Web Conference (ISWC '04). Springer, Berlin Heidelberg, pp 244–258
44. Manca M, Fabio P, Santoro C, Corcella L (2019) Supporting end-user debugging of trigger-action rules for IoT applications. International Journal of Human-Computer Studies 123:56–69
45. Markopoulos P, Nichols J, Paternò F, Pipek V (2017) Editorial: end-user development for the Internet of Things. ACM Transactions on Computer-Human Interaction 24(2):1–3
46. Metaxas G, Markopoulos P (2017) Natural contextual reasoning for end users. *ACM Transaction on Computer-Human Interaction* 24(2) Article 13:1–36
47. Microsoft. *Microsoft Flow*. Retrieved from https://flow.microsoft.com/. Accessed 28 Feb 2020

48. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
49. Mohanraj I, Ashokumar K, Naren J (2016) Field monitoring and automation using IOT in agriculture domain. Procedia Computer Science 93:931–939
50. Mora S, Gianni F, Divitini M (2017) Tiles: a card-based ideation toolkit for the Internet of Things. In Proc. of the Conference on Designing Interactive Systems (DIS '17). ACM, New York, pp 587–598
51. Noy NF, Fergerson RW, Musen MA (2000) The knowledge model of Protégé-2000: combining interoperability and flexibility. In *Proc. of the International Conference on Knowledge Engineering and Knowledge Management (EKAW '00)*. Springer Berlin Heidelberg, 17–32
52. Oppezzo M, Schwartz DL (2014) Give your ideas some legs: the positive effect of walking on creative thinking. *Journal of Experimental Psychology: Learning, Memory, and Cognition (PeePs)* 40(4):1142
53. Patil AA, Oundhakar SA, Sheth AP, Verma K (2004) Meteor-s web service annotation framework. In Proc. of the International Conference on World Wide Web (WWW '04). ACM, New York, pp 553–562
54. Petrelli D, Lechner M (2014) The meSch project – material EncounterS with digital cultural heritage: reusing existing digital resources in the creation of novel forms of visitor's experiences. In Proc. of the International Committee for Documentation of ICOM (CIDOC '14)
55. Petrelli D, Marshall MT, O'Brien S, McEntaggart P, Gwilt I (2017) Tangible data souvenirs as a bridge between a physical museum visit and online digital experience. Pers Ubiquit Comput 21(2):281–295
56. Plaisant C, Grosjean J, Bederson BB (2002). SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proc. of the IEEE Symposium on Information Visualization (INFOVIS 2002)*, 57–64
57. Gennari R, Matera M, Melonio A, Roumelioti E, Delprino F, Filaferro M, Gennaioli G, Manzi V(2019). Interactive nature: designing smart devices for nature exploration by children. In *Proc. of the Biannual Conference of the Italian Chapter of SIGCHI 2019 - Adjunct Proceedings, CLEUP (CHItaly '19)*, 67–70
58. Robertson, G.G., Mackinlay, J.D. and Card, S.K. (1991). Cone trees: animated 3D visualizations of hierarchical information. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*. ACM, New York, pp 189–194
59. Rodríguez-García MÁ, Valencia-García R, García-Sánchez F, Samper-Zapater JJ (2014) Ontology-based annotation and retrieval of services in the cloud. Knowl-Based Syst 56:15–25
60. Rosson MB, Carroll JM (2003) Scenario-based design. In: Julie AJ, Andrew S (eds) *The human-computer interaction handbook*. L. Erlbaum associates Inc., Mahwah, pp 1032–1050
61. Saquicela V, Vilches-Blázquez LM, Corcho Ó (2010). Semantic annotation of RESTful services using external resources. In *Proc. of the International Conference on Web Engineering (ICWE '10)*. Springer Berlin Heidelberg, pp 266–276
62. Shneiderman B (2007) Creativity support tools: accelerating discovery and innovation. Commun ACM 50(12):20–32
63. Shneiderman B (2009) Creativity support tools: a grand challenge for HCI researchers. In: Redondo M, Bravo C, Ortega M (eds) Engineering the user interface: from research to practice. Springer London, London, pp 1–9
64. Sivakumar R, Arivoli P, Sri A (2011) Ontology visualization PROTÉGÉ tools–a review. International Journal of Advanced Information Technology 1(4):1–11
65. Sure Y, Angele J, Staab S (2002). OntoEdit: guiding ontology development by methodology and inferencing. In *Proc. of the OTM Confederated International Conferences "On the Move to Meaningful Internet Systems" (OTM '02)*. Springer Berlin Heidelberg, 1205–1222
66. Tutenel T, Bidarra R, Smelik RM, Kraker KJD (2008) The role of semantics in games and simulations. Comput Entertain 6(4):1–35
67. Unger R, Chandler C (2012). A project guide to UX design: for user experience designers in the field or in the making. New Riders
68. Ur B, Ho MPY, Brawner S, Lee J, Mennicken S, Picard N, Schulze D, Littman ML (2016). Trigger-action programming in the wild: an analysis of 200,000 IFTTT recipes. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, pp 3227–3231
69. Warr A, O'Neill E (2005). Understanding design as a social creative process. In *Proc. of the Conference on Creativity & cognition (C&C '05)*. ACM, New York, pp 118–127
70. Zancanaro M, Not E, Petrelli D, Marshall M, van Dijk T, Risseeuw M, van Dijk D, Venturini A, Cavada D, Kubitza T (2015). Recipes for tangible and embodied visit experiences. In *Proc. of the Museums and the Web conference (MW '15)*
71. Zhong S, Tian L, Li C, Storch K-F, Wong WH (2004). Comparative analysis of gene sets in the gene ontology space under the multiple hypothesis testing framework. In *Proc. of the IEEE Computational Systems Bioinformatics (CSB '04)*. IEEE, 425–435
72. Zobel J, Dart P (1996). Phonetic string matching: lessons from information retrieval. In *Proc. of the ACM SIGIR conference on Research and development in information retrieval (SIGIR '96)*. ACM, New York, pp 166–172