

Specification of Complex Logical Expressions for Task Automation: An EUD Approach

Giuseppe Desolda^{1(✉)}, Carmelo Ardito¹, and Maristella Matera²

¹ Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro,
Via Orabona, 70125 Bari, Italy

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

² Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano,
Piazza Leonardo da Vinci 32, 20134 Milan, Italy

maristella.matera@polimi.it

Abstract. The growing availability of smart objects is stimulating researchers in investigating the IoT phenomenon from different perspectives. In the HCI area, and in particular from the EUD perspective, one prominent goal is to enable non-technical users to be directly involved in configuring smart object behaviour. With this respect, this paper discusses three visual composition techniques to specify logical expressions in Event-Condition-Action rules used for synchronizing the behavior of smart objects.

Keywords: End User Development · Internet of Things · Visual languages · Logical expressions

1 Introduction and Motivation

In the last years, the computer science landscape has been shifting from the use of personal computer and mobile devices to the adoption in several contexts of integrated, powerful, pervasive devices that offer features like connectivity, sensors, actuators and embedded software, the so-called “smart objects” [5]. The possibility to create easily low-cost smart objects has resulted in the phenomenon known as Internet of Things (IoT), a term indicating a computing platform where the physical world is connected to the Internet via ubiquitous smart objects.

So far, research on IoT has primarily focused on the technological aspects characterizing smart objects; instead, few works have concentrated on how to amplify the social and practical benefits for the end users. We believe that the potential of IoT can be increased if even users with no expertise in computer programming are enabled to manage interoperable smart objects to respond to their situational needs [6].

Smart objects are typically provided with software services (e.g., mobile apps) to interact with them. Unfortunately, it is not possible to go beyond predefined behaviors. For example, a smart bracelet has sensors that can track sleep cycles, heart rate and steps. The measured values are accessible by users through a specific mobile app. However, no communication is allowed between the bracelet and other smart objects. This communication would instead enable the definition of composite behaviors, such as

controlling the opening of a smart roll-up shutter when the smart bracelet detects that the user is awake. New solutions are needed to allow non-technical users to orchestrate ecologies of smart objects to satisfy the “long tail” of specific – and sometimes unexpected – needs [9, 14]. In other words, adequate End-User Development (EUD) paradigms are needed. Indeed, as largely recognized in the literature [3, 8, 15, 22], EUD methodologies fit very well the requirement of letting users customize their systems to support personal, situational needs.

Several Web tools already address the problem of Task Automation (TA) by supporting the creation of Event-Condition-Action (ECA) rules to synchronize the behavior of smart objects [7, 16]. However, their composition paradigms for rule creation are still far from offering evident benefits for the end users [6]. There are tools [17–19, 23] that claim to address the skills of non-technical users, but they only assist the creation of “basic” rules, for example synchronizing one single event with one single action. Some other tools (e.g., Node-RED [21]) allow one to create more expressive rules, but they also require programming skills.

In this paper we present three novel visual composition techniques for the definition of logical expressions during the creation of Task-Automation rules. The first technique proposes abstraction mechanisms to combine rule events by means of AND/OR logical operators, as well as to group set of conjunctive/disjunctive events, also recursively. The second technique constrains the creation of logic expressions taking into account a principle of the mental model theory [20] saying that people find easier the conceptualization of logical statements as a disjunction of conjunctions (Disjunctive Normal Form - DNF). The third technique is the opposite of DNF, since it allows the combination of rule events as a *conjunction of disjunction* (Conjunctive Normal Form - CNF). Even if the mental model theory prefers the DNF, we believe that CNF can also be a valid support to specify ECA rules.

The paper is organized as follows. Section 2 describes EFESTO-5 W, a Web platform implementing a composition paradigm for defining easily Task-Automation rules. Section 3 then describes the new extensions introduced in EFESTO-5 W to increase the expressive power of logical formulas; it also discusses some works that guided the definition of the new composition mechanisms. Finally, Sect. 4 concludes the paper and outlines our future work.

2 EFESTO-5W: A Platform to Create Task-Automation Rules

The research we have conducted in the last years has primarily focused on the definition of composition paradigms assisting non-technical end users in the composition of Web services [2, 4, 10, 11]. Given the diffusion of smart objects that typically expose their sensors and actuators as Web services, we recently started investigating how end users can be enabled to synchronize interoperable smart objects and Web services, to respond to their situational needs [13]. As result, we designed and developed EFESTO-5W, a Task-Automation tool for non-technical users. Its initial design was driven by the 5W model, typically adopted in journalism to analyze the complete story about a fact by

answering to the 5 questions: (1) *Who* did it? (2) *What* happened? (3) *When* did it take place? (4) *Where* did it take place? (5) *Why* did it happen?

These five questions were used during an elicitation study to guide participants in proposing user interfaces and interaction techniques for the creation of ECA rules. The design phase resulted in three different user interfaces and interaction paradigms that we then compared during a rigorous study to identify the best one [12].

In order to describe the EFESTO-5W composition paradigm, we illustrate a usage scenario. A user, Jaime, wants to create the following rule: the garage door automatically opens and the smart boiler turns on when his car is close to home and when he taps his smart watch. To start creating this rule, in the visual interface illustrated in Fig. 1 (initially not showing any event and action) Jaime clicks the “New Rule” button (Fig. 1, circle 1) and the “Creating Rule” window is opened. This is the main composition area in which a rule is going to be defined. The left side is for specifying the triggering events, and the right side is to define the actions to be activated.

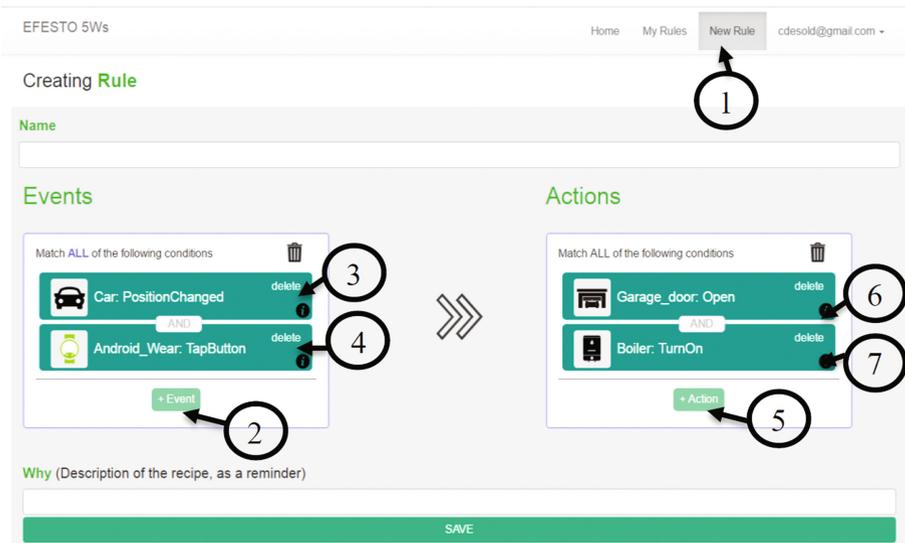


Fig. 1. EFESTO-5 W interface for rule creation. (Color figure online)

A wizard procedure, activated by the green “+Event” button (Fig. 1, circle 2), guides Jaime in defining the rule events. The wizard sequentially shows some pop-up windows to select and configure the different elements composing a rule. Jaime first defines the events in a WYSIWYG manner, in terms of *Which* is the service to be monitored for detecting the triggering events, *What* service events have to be monitored, *When* and *Where* the events have to occur. The specification of *When* and *Where* conditions is optional. Thus, following the wizard procedure, Jaime initially selects the *Car* object (Which) and the *Position changed* event (What). He also constraints the event occurrence in the time interval 7.00 p.m.–10.00 p.m. (When) and in proximity of his home address (Where). At the end of the wizard procedure, the summary of the defined event

appears in the “Events” area (Fig. 1, circle 3). In a similar way, Jaime adds another event that, in order to activate the rule actions, has to occur together with the first one. He thus selects the *smartwatch* (Which) and its *Tap* event (What) (Fig. 1, circle 4).

The two defined events will trigger the rule as soon as both of them occur. This is expressed by the sentence “Match ALL of the following conditions”, on top of the Events blocks, that indicates that all the specified events will be combined by the AND logical operator. The user can click on the “ALL” word (in blue) to switch to the modality “Match AT LEAST ONE of the following conditions”, which means that the occurrence of just one event is needed in order to activate the rule; in this case, indeed, the events are combined by the OR logical operator.

After defining the events, Jaime starts adding the actions to be executed in response to the specified events. He clicks on the “+Action” button (Fig. 1, circle 5); a wizard procedure starts, which is similar to the one used for the events. Jaime chooses the *Garage Door* (Which) and its *Open* action (What) (Fig. 1, circle 3). Afterwards, he repeats the same procedure by selecting the *Boiler* (Which) and its *Turn on* action (What) (Fig. 1, circle 4). At the end, he saves the rule; the rule is now active.

3 Complex Logical Expressions for Task-Automation Rules

As illustrated in the previous Section, in the first version of EFESTO-5 W users could specify multiple events and actions, but they were allowed to use only one single logical operator to combine rule events: *AND*, if they wanted to activate the rule actions once *all the events* occurred; *OR*, if they wanted to activate the rule actions when *at least one event* occurred. We constrained the rule definition mechanisms in such a way because in the conducted user studies we wanted to focus on the specification of temporal and spatial constraints [12, 13]. Therefore, we purposely did not introduce further elements that could generate side effects deviating from the factors that we wanted to investigate. In the following, we illustrate the extensions of the visual composition paradigm recently introduced to support the combination of multiple events by means of arbitrarily expressive logical expressions.

3.1 Logical Expressions in Task-Automation Rules

It is widely recognized that Boolean algebra is a formalism for describing logical relations in the same way as ordinary algebra describes numeric relations. In particular, variables can be combined by using *conjunction*, *disjunction* and *negation* operators; the values of the variables are the truth values *true* and *false*. Such formalism fits the mental model of computer scientists and engineers, while non-technical users might not be able to use them properly. Often non-technical users tend to interpret “or” as an exclusive OR. Sometimes they have problems managing nested parentheses [20]. In the following we describe some extensions of the EFESTO-5W composition paradigm that aim to alleviate such problems.

As starting point of our design, we surveyed the literature and identified systems that support the definition of logical expressions by using visual techniques. One traditional

approach is to represent logic operators graphically with a plumbing or electrical metaphor [24, 26]. However, in the context of visual composition of Web services, these interaction metaphors did not succeed because they recall concepts that are closer to the background of ICT experts and electronic engineers than to the end users' mental model [25, 27]. Alternative composition paradigms are offered by some commercial and open source systems. For example, Apple *iTunes 9* supports the creation of smart playlists with nested conditionals¹, in order to play songs that match settings within user-defined rules. Apple has also implemented a mechanism inside *Mail*, for creating rules that for example automatize the management of incoming messages, as well as the replying to or the forwarding of messages². A recent open-source project, *QueryBuilder* [1], also provides a Web-based user interface to visually create queries and filters.

Inspired by these approaches, our team of three HCI experts carried out a user-centered design, involving four non-technical end users, to identify alternative visual composition techniques for combining smart-object events by means of logical expressions. As a first result, we identified a technique to combine the rule events with the AND/OR logical operators, as well as to group set of conjunctive/disjunctive events, also recursively. With respect to the EFESTO-5W interface shown in Fig. 1, in the Events area we introduced a root-block with the label "Match AT LEAST ONE of the following conditions" (the white block in Fig. 2). This label was conceived to express in natural language the semantics of the OR operator. Inside the white block there are two buttons, i.e., "+Event" and "+Block of Events Conjunction" (see Fig. 2). The first one allows the user to define events as in the previous scenario. The second one permits the creation of sub-blocks grouping set of conjunctive events and is titled "Match ALL of the following conditions" (this label expresses in natural language the semantics of the AND operator). The rule actions are activated when at least one of the events or one of the sub-blocks inside the root are evaluated as true³.



Fig. 2. Starting block to define events as logical expressions. (Color figure online)

Inside each sub-block the users can recursively define events or block of events. The logical connectors of the root block and of all the sub-blocks can be switched between OR/AND by clicking on the related label ("Match AT LEAST ONE of the following conditions" and "Match AT LEAST ONE of the following conditions").

The users can exploit this technique to compose different events. For example, the rule depicted in Fig. 3 refers to a scenario where a user, Giuseppe, wants to optimize the home energy consumption by automatically turning on the house heater at 25 °C and

¹ http://www.macworld.com/article/1142846/nested_playlists.html.

² <http://www.macworld.com/article/1159394/business/mailrules.html>.

³ EFESTO-5 W checks events every N minutes, being N a parameter that can be set by the user.

turning on the boiler when one of the following conditions are true: (1) his car (*Giuseppe_Car*) is close to home and he taps his *smartwatch*; (2) his wife car (*Lucy_Car*) is close to home and his wife taps her *smartwatch*; (3) his smart bracelet detects that he is just awake.

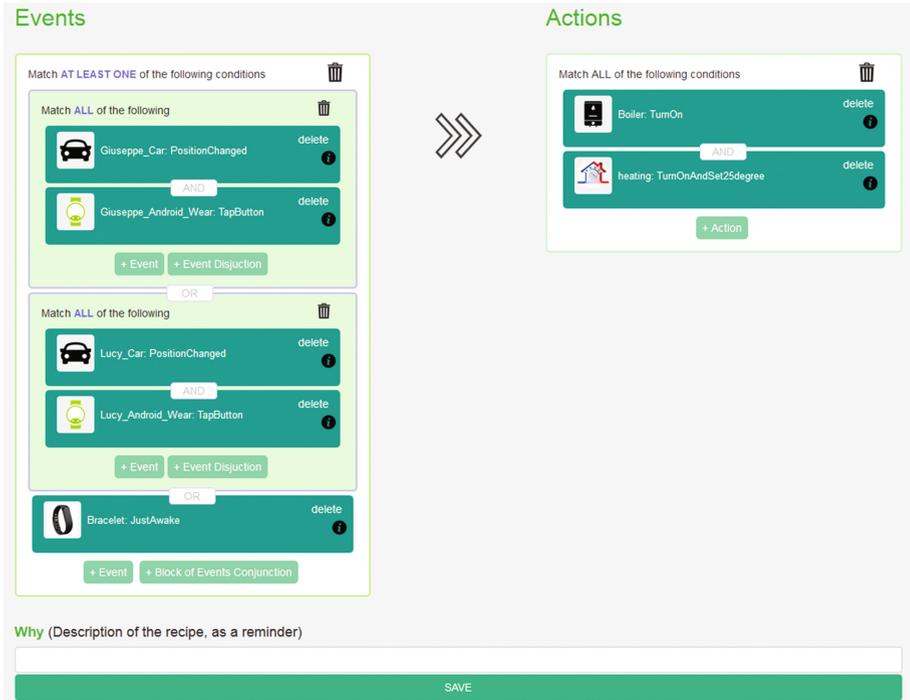


Fig. 3. Example of rule with multiple events.

3.2 Simplifying the Creation of Logical Expressions

Even if the visual technique described above introduces abstraction mechanisms that facilitate the creation of logical expression, we designed two further solutions based on the disjunctive normal form (DNF) and the conjunctive normal form (CNF). A DNF is a normalization of a logical formula that is a disjunction of clauses that themselves can be simple terms or a conjunction of terms (terms are smart object events in our case). Contrarily, a CNF is a normalization of a logical formula that is a conjunction of clauses that themselves can be a term or a disjunction of terms. Every propositional formula can be converted into an equivalent DNF and CNF formula. This transformation is based on logical equivalences, such as the double negative law, De Morgan's laws, and the distributive law.

According to the mental model theory [20], people find easier the conceptualization of logical statements as a disjunction of conjunctions. Thus, we designed a DNF-based technique. With respect to Fig. 3, the root-block is titled "Match AT LEAST ONE of

the following conditions” and contains two buttons: “+Event” and “+Block of Events Conjunction”. Contrarily to what the mental model theory predicts, we believe that also a CNF-based technique can simplify the creation of logical expressions when composing smart objects. With respect to Fig. 3, the root-block is titled “Match ALL of the following conditions” and contains two buttons: “+Event” and “+Block of Events Disjunction”. In both the DNF-based and CNF-based techniques, users cannot switch the logical operators like in the previous example and cannot recursively create sub-blocks. Lack of space prevents us to show pictures of DNF and CNF techniques. However, a video at the link <https://goo.gl/dn1bde> shows the three visual composition techniques.

4 Conclusion and Future Work

This paper has presented some new extensions of the EFESTO-5W composition paradigm that aim to grant more expressivity in the specification of Task-Automation rules by end users without any expertise in programming. The new techniques were defined taking into account some findings reported in the literature about factors that can increase the attitude of non-technical users towards expressing logical expressions. We also took into account the feedback of a group of non-technical users involved in the design activities.

The previous version of the composition paradigm was extensively evaluated and compared with other paradigms adopting different interaction metaphors [12]. Through a family of studies, we identified some key factors of composition paradigms that can improve the performance and satisfaction of non-technical users. The new extensions do not contradict such factors. However, we need to verify whether the new level of complexity (e.g., understanding how to use different logical operators) would degrade the ease-of-use and the intuitiveness of the composition paradigm. We have already designed some user studies to evaluate the effects implied by the new extensions. We will then investigate how to extend the use of both AND and OR operators also to the specification of combination of actions.

The overall goal of the research will be to devise an EUD environment where ecosystems of smart objects can be represented through adequate metaphors and easily manipulated and controlled by end users. We are already designing such environment for the definition of smart spaces in museums. From the experience in this specific domain we will also try to derive more general hints for the design of EUD environments for the Internet of Things.

References

1. StrangePlanet. QueryBuilder (2017). <http://querybuilder.js.org/>
2. Ardito, C., Bottoni, P., Costabile, M.F., Desolda, G., Matera, M., Picozzi, M.: Creation and use of service-based distributed interactive workspaces. *J. Vis. Lang. Comput.* **25**(6), 717–726 (2014)
3. Ardito, C., Buono, P., Costabile, M.F., Lanzilotti, R., Piccinno, A.: End users as co-designers of their own tools and products. *J. Vis. Lang. Comput.* **23**(2), 78–90 (2012)

4. Ardito, C., Costabile, M.F., Desolda, G., Lanzilotti, R., Matera, M., Picozzi, M.: Visual composition of data sources by end-users. In: Proceedings of the International Conference on Advanced Visual Interfaces (AVI 2014), Como (Italy), 28-30 May, pp. 257–260. ACM, New York (2014)
5. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. *Int. J. Comput. Comput. Netw.* **54**(15), 2787–2805 (2010)
6. Barricelli, B.R., Valtolina, S.: Designing for end-user development in the internet of things. In: Díaz, P., Pipek, V., Ardito, C., Jensen, C., Aedo, I., Boden, A. (eds.) IS-EUD 2015. LNCS, vol. 9083, pp. 9–24. Springer, Cham (2015). doi:[10.1007/978-3-319-18425-8_2](https://doi.org/10.1007/978-3-319-18425-8_2)
7. Coronado, M., Iglesias, C.A.: Task automation services: automation for the masses. *IEEE Internet Comput.* **20**(1), 52–58 (2016)
8. Costabile, M.F., Fogli, D., Mussio, P., Piccinno, A.: Visual interactive systems for end-user development: a model-based design methodology. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **37**(6), 1029–1046 (2007)
9. Daniel, F., Matera, M., Weiss, M.: Next in mashup development: user-created apps on the web. *IT Prof. Magaz.* **13**(5), 22 (2011)
10. Desolda, G.: Enhancing workspace composition by exploiting linked open data as a polymorphic data source. In: Damiani, E., Howlett, Robert J., Jain, L.C., Gallo, L., De Pietro, G. (eds.) Intelligent Interactive Multimedia Systems and Services. SIST, vol. 40, pp. 97–108. Springer, Cham (2015). doi:[10.1007/978-3-319-19830-9_9](https://doi.org/10.1007/978-3-319-19830-9_9)
11. Desolda, G., Ardito, C., Matera, M.: EFESTO: a platform for the end-user development of interactive workspaces for data exploration. In: Daniel, F., Pautasso, C. (eds.) RMC 2015. CCIS, vol. 591, pp. 63–81. Springer, Cham (2016). doi:[10.1007/978-3-319-28727-0_5](https://doi.org/10.1007/978-3-319-28727-0_5)
12. Desolda, G., Ardito, C., Matera, M.: Empowering end users to customize their smart environments: model, composition paradigms and domain-specific tools. *ACM Trans. Comput. Hum. Interact. (TOCHI)* **24**(2), 53 (2017). Article 12
13. Desolda, G., Ardito, C., Matera, M.: End-user development for the internet of things: EFESTO and the 5W composition paradigm. In: Daniel, F., Gaedke, M. (eds.) RMC 2016. CCIS, vol. 696, pp. 74–93. Springer, Cham (2017). doi:[10.1007/978-3-319-53174-8_5](https://doi.org/10.1007/978-3-319-53174-8_5)
14. Fischer, G.: End-user development and meta-design: foundations for cultures of participation. In: Pipek, V., Rosson, M.B., Ruyter, B., Wulf, V. (eds.) IS-EUD 2009. LNCS, vol. 5435, pp. 3–14. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-00427-8_1](https://doi.org/10.1007/978-3-642-00427-8_1)
15. Fischer, G., Giacardi, E., Ye, Y., Sutcliffe, A., Mehandjiev, N.: Meta-design: a manifesto for end-user development. *Commun. ACM* **47**(9), 33–37 (2004)
16. Fogli, D., Lanzilotti, R., Piccinno, A.: End-user development tools for the smart home: a systematic literature review. In: Streitz, N., Markopoulos, P. (eds.) Distributed, Ambient and Pervasive Interactions, in DAPI 2016. LNCS, vol. 9749, pp. 69–79. Springer, Cham (2016)
17. elastic.io GMBH (2016). <http://www.elastic.io/>
18. IFTTT (2016). <https://ifttt.com/>
19. Zapier Inc. Zapier (2016). <https://zapier.com/>
20. Johnson-Laird, P.N.: Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness. Harvard University Press, Cambridge (1983)
21. JS_Foundation. Node-RED (2016). <http://nodered.org/>
22. Lieberman, H., Paternò, F., Wulf, V. (eds.) End User Development. Human–Computer Interaction Series, vol. 9. Springer, Dordrecht (2006)
23. Itrios LLC. itDuzzit (2016). <http://cloud.itduzzit.com/>
24. Murray, N.S., Paton, N.W., Goble, C.A., Bryce, J.: Kaleidoquery—a flow-based visual language and its evaluation. *J. Vis. Lang. Comput.* **11**(2), 151–189 (2000)

25. Namoun, A., Nestler, T., Angeli, A.: Conceptual and usability issues in the composable web of software services. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 396–407. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-16985-4_35](https://doi.org/10.1007/978-3-642-16985-4_35)
26. Shneiderman, B.: Visual user interfaces for information exploration. In: Proceedings of the 54th Annual Meeting of the American Society for Information Sciences (ASIS 1991), pp. 379–384 (1991)
27. Wajid, U., Namoun, A., Mehandjiev, N.: Alternative representations for end user composition of service-based systems. In: Costabile, M.F., Dittrich, Y., Fischer, G., Piccinno, A. (eds.) IS-EUD 2011. LNCS, vol. 6654, pp. 53–66. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-21530-8_6](https://doi.org/10.1007/978-3-642-21530-8_6)