A User-Friendly Interface for Rules Composition in Intelligent Environments

A User-Friendly Interface for Rules Composition in Intelligent Environments

Dario Bonino, Fulvio Corno, Luigi De Russis

Abstract In the domain of rule-based automation and intelligence most efforts concentrate on building the technological infrastructure, often disregarding user-home interaction requirements. This paper attempts to mitigate this issue by defining a rich-web rule visual design interface specifically aimed at non-skilled home inhabitants.

Key words: Domotics, Rules Composition, Intelligent Environment, Human-Home Interaction, User Interface, Smart Home, Interface Concept

1 Introduction

Many intriguing scenarios are currently sketching the home of the future, where human inhabitants will only carry out "exciting" or "interesting" tasks and the home will take care of all boring duties that fill every day life. Although appealing, this long-term vision (part of the Ambient Intelligence research field) has also a worrying connotation where homes not only facilitate our life but directly modify our home-related behavior in ways difficult to forecast, on the user side. This scenario, already emerging from several studies about user attitudes towards smart homes [1, 3], has been driving an initial research effort on finding suitable trade-offs between totally direct user control and fully automatic home behaviors, involving several degrees of home autonomy.

No sound and widely agreed solution to this trade-off has currently been found and the related research activities, both in the Human Computer Interaction (HCI) and Ambient Intelligence (AmI) communities are still very active. Nevertheless,

Dario Bonino, Fulvio Corno, Luigi De Russis
Politecnico di Torino, Dipartimento di Automatica ed Informatica, Corso Duca degli Abruzzi 24, 10129 - Torino, Italy, e-mail: {dario.bonino, fulvio.corno, luigi.derussis}@polito.it

a relatively accepted approach based on *activity delegation* is gaining momentum. In the AmI community, explicit delegation of tasks to homes is usually realized through rule-definition or user-initiated learning. While technology, especially for rule-based delegation, is rather mature and widely investigated, there is still a sensible lack of effective user interfaces. To support users in shaping their specific home automation policies, interfaces must be simple, easy to use and to learn for people without advanced computer skills, and should not require any specific notion about the automation technology installed in the home.

In this paper we propose a first step for overcoming the current lack of effective rule definition interfaces by proposing a paper prototype of a rich-web rule design interface specifically aimed at non-skilled home inhabitants. The remainder of the paper is organized as follows: Section 2 relates the proposed approach to the state of the art, highlighting commonalities and differences. Section 3 defines requirements for rule creation interfaces dedicated to non-skilled home inhabitants, while Section 4 introduces the interface design. Section 5 concludes the paper and depicts future works and research scenarios.

2 Related Works

Different rule composition interfaces, aimed at people without technical skills, are present in the literature, either for domestic environments or for other application domains. An example of such interfaces is OpenBlocks. *OpenBlocks* [4] is a general-purpose framework for graphical block programming systems, developed at the MIT. This framework is more expressive than the interface we propose and could represent a complementary solution to realize a rule builder. However, OpenBlocks requires a heavy customization to be adapted for a domotic environment, it is not web-based and its higher expressiveness leads to increased complexity for rules creation.

Most of rule builder applications specifically developed for domestic environment are *context-aware* applications, such as iCAP [2]. The *iCAP* interface has one window with two main areas. A tabbed window is the repository for user-defined inputs, outputs, and rules. The input and output components are associated with graphical icons that can be dragged into the main area, to be later used to construct a conditional rule statement. An example rule could be: *IF Sam is in the office after 5pm and the temperature is less than 10 Celsius degrees OR IF Jane is in the bedroom and the temperature is between 0 and 15 Celsius degrees, THEN turn on the heater in the house*. The grammar used by iCAP is similar to the one proposed by our interface, but iCAP is not web-based, requires the user to draw each object she wants to use in a rule, is pen-based and does not differentiate between events and conditions.

3 Requirements

Delegating part of everyday tasks to the home requires suitable interfaces for enabling the home inhabitants to easily define processes to be automated, i.e., to effectively program automation rules. By interacting with both people living and managing smart homes and with people commercializing wired and wireless home automation systems we derived the following set of requirements that an effective rule builder interface shall obey.

1. Rules shall be ***definable by people with basic level of computer literacy***, the only required knowledge is about the home components, in terms of normal usage and behavior.
   a. Home *devices shall be exposed in an abstract and technology independent way*, thus enabling user to easily specify the rule objects.
   b. *Rules shall be self-explaining*, i.e., they can be directly/easily translated in a nearly natural language description.
   c. *Rules shall always be "valid"*, i.e., the user can only create and save syntactically (and possibly semantically) correct rules.
   d. Rules shall be *expressive enough to manage most situations, actions and interactions* that a home inhabitant may want to delegate (i.e., they shall be easily mapped onto a powerful enough computer-based rule language).
2. Rules shall be defined by using a ***wide range*** of possible ***interface devices*** (e.g., PCs, touch screen panels), and ***input modalities*** (e.g., touch, mouse).
3. The ***rule-design interface must facilitate*** the ***delegation of tasks*** from humans to homes ***providing suitable "aids"***.
   a. Rules editing shall be facilitated by means of *suggestions, guiding interfaces* and *auto-filling* functionalities.
   b. Rule interface should offer *support to handle unexpected loss of connections* or *computer malfunctions*, e.g., *automatically saving rules*.

4 Design

Interface Concept
Sam is a smart home user with little technological skills. Thanks to a web application, he configured the devices present in his home, renaming them at his pleasure. Now, he wants to create a rule such as "*if the living room is dark, turn on the lamp*". Sam opens his browser to reach the Rule Builder. The interface he sees on his screen is sketched in Figure 1. On the left, he sees what he needs to create the rule, including a lamp and a light sensor, previously added from the house map. On the right, he can see a wide area to be used for the definition of a rule. The dotted rectangles under the IF and THEN keywords are strong visual clues suggesting to drag a device inside them (req. 3a). Sam decides to drag the light intensity sensor under the

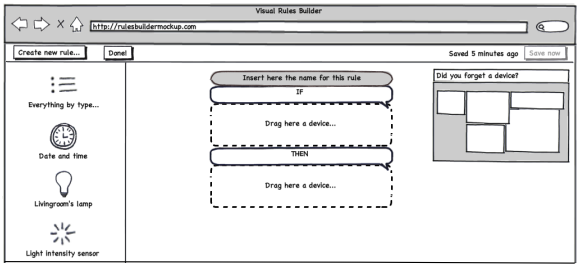

Fig. 1 The main page of our Rule Builder

"IF". By dragging the icon, it docks under the "IF" as a rectangular container. In this container, besides the sensor name, Sam sees a list to specify what sensor event should be intercepted. He chooses "LOW" light intensity (req. 1a).

Sam, after, drags the lamp icon under the "THEN" keyword. When he starts to drag the icon, two other rectangles appear before the "THEN" keyword (req. 3a and 1c): the optional "WHEN" and "OR IF" statements (req. 1b and 1d). He continues to drag the lamp icon under the "THEN" keyword and selects "ON" between the options presented by the lamp container (req. 1a).

The rule is complete (Figure 2) and the lower part of the interface reports a sentence that summarizes the just created rule (req. 1b).

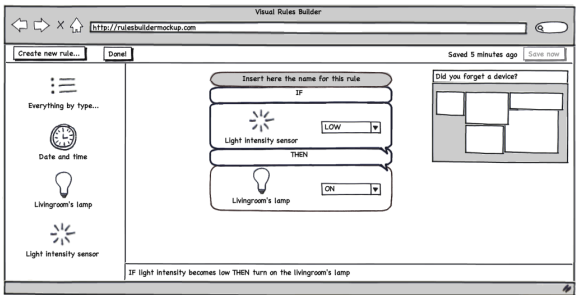

Fig. 2 Complete rule

Grammar

The Rules Builder concept just illustrated guarantees rule correctness (req. 1c) and readability (reqs. 1b, 1d and 3a) by exploiting a formal rule representation grammar (see Figure 3) based upon four fixed keywords: IF, THEN, WHEN, OR IF (req. 1c and 1d). The first two are mandatory for the creation of any rule, while the others are optional (dotted in Figure 3). A rule composed with this grammar follows the natural language (req. 1b). The IF keyword expresses an event to trigger the rule. The event is indicated, in Figure 3, as an "E-BLOCK" (event-block). WHEN defines one or more conditions constraining the event; multiple constraints should be simultaneously satisfied. The set of constraints is shown as "C-BLOCKS"

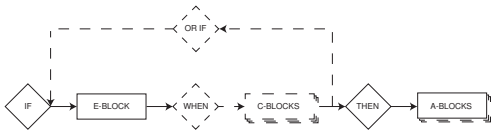

Fig. 3 The grammar underlying the creation of a rule

(constraint-blocks). OR IF is a disjunction for repeating the IF-WHEN part more than once. Finally, THEN indicates a set of actions to be executed on the occurrence of the above triggers. The actions are indicated as "A-BLOCKS" (action-blocks).

To maintain rule consistency (req. 1c), each device involved in the creation of a rule has a different behavior according to the block in which is inserted: E-BLOCK interprets events generated by controllable devices, clock and sensors; C-BLOCK supports controllable devices, clock and sensors; A-BLOCK, finally, supports controllable devices only.

The interface concept and the Rules Builder grammar have been informally verified, and approved, by a restricted test group with basic computer skills.

5 Conclusions and Future Works

This paper distills the basic requirements of home rule development environments and introduces the conceptual design of Rule Builder, a rich-web interface that specifically targets non-expert users, i.e., home inhabitants with little or no technological skills. Future works are the realization of a working prototype fulfilling all the requirements proposed and its integration with Dog, an ontology-based domotic gateway[1]. Extensive experimentation with users will then provide means for a sound validation of the proposed interface.

References

1. Demiris, G., Rantz, M., Aud, M., Marek, K., Tyrer, H., Skubic, M., Hussam, A.: Older adults' attitudes towards and perceptions of "smart home" technologies: a pilot study. Medical Informatics and the Internet in Medicine **29(2)**, 87–94 (2004)
2. Dey, A., Sohn, T., Streng, S., Kodama, J.: iCAP: Interactive Prototyping of Context-Aware Applications. In: K. Fishkin, B. Schiele, P. Nixon, A. Quigley (eds.) Pervasive Computing, *Lecture Notes in Computer Science*, vol. 3968, pp. 254–271. Springer Berlin / Heidelberg (2006)
3. Eggen, B., Hollemans, G., van de Sluis, R.: Exploring and Enhancing the Home Experience. Cognition, Technology and Work **Volume 5, Number 1 / April, 2003**, 44–54 (2003)
4. Roque, R.V.: OpenBlocks : an extendable framework for graphical block programming systems. Massachusetts Institute of Technology (2007). URL `http://hdl.handle.net/1721.1/41550`

[1] `http://domoticdog.sourceforge.net`