POLITECNICO DI TORINO
Repository ISTITUZIONALE

HomeRules: A Tangible End-User Programming Interface for Smart Homes

Original
HomeRules: A Tangible End-User Programming Interface for Smart Homes / DE RUSSIS, L., Corno, F.. - (2015), pp. 2109-2114. (CHI'15: CHI Conference on Human Factors in Computing Systems Seoul, Republic of Korea April 18 - 23, 2015) [10.1145/2702613.2732795].

Availability:
This version is available at: 11583/2588215 since:

Publisher:
ACM

Published
DOI:10.1145/2702613.2732795

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

20 November 2026

HomeRules: A Tangible End-User Programming Interface for Smart Homes

Luigi De Russis
Politecnico di Torino
Corso Duca degli Abruzzi, 54
Torino, 10129 Italy
luigi.derussis@polito.it

Fulvio Corno
Politecnico di Torino
Corso Duca degli Abruzzi, 54
Torino, 10129 Italy
fulvio.corno@polito.it

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
CHI'15 Extended Abstracts, Apr 18-23, 2015, Seoul, Republic of Korea
ACM 978-1-4503-3146-3/15/04.
http://dx.doi.org/10.1145/2702613.2732795

Abstract
A considerable amount of research has been carried out towards enabling average users to customize their smart homes through trigger-action ("if... then...") programming. However, inhabitants of such smart environments keep having problems understanding, administering, troubleshooting, and deriving benefits from the technologies employed in their homes. By synthesizing a broad body of research on end-user programming in smart homes with observations of commercial products and our own experiences, we provide a set of guidelines for designers of future interfaces and tools. Stemming from them, we present the design and the initial evaluation of HomeRules, a mobile and tangible application for end-user programming in smart homes.

Author Keywords
Smart home; home automation; end-user programming; mobile programming; guidelines

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

Introduction
In the last years, smart homes gained a new momentum, thanks to an increased availability of commercial solutions

and to steadily reducing costs. As a result, a new body of knowledge is now available to the research community and to the industry, and further explorations "in the wild" are finally possible.

Mennicken et al., reporting the results from a qualitative study on various stakeholders of smart home technologies [4], discovered that home inhabitants, particularly those who do not have any technical knowledge, have problems understanding, administering and troubleshooting smart home technologies. This means that a true collaboration between the home and their inhabitants is still unreached: in fact, autonomous technologies often leave users feeling out of control, without the possibility to adjust the level of autonomy of their home according to their needs [1]. To avoid this feeling, researchers and practitioners are currently proposing interfaces and tools that allow home inhabitants to explicitly delegate specific tasks, maybe boring or repetitive, to their smart homes.

A promising approach to delegate such tasks and create connections between individual devices seems to be to express the inhabitants needs, wishes and desiderata in a more natural, less technical form [5], trying to avoid using the vocabulary and the typical metaphors adopted by and for people with technical background. In this way, an inhabitant-centered approach is ensured: users are always in the loop and perceive the smart home as something both convenient and useful. Moreover, according to Newman [6], end-user configurability is a key factor in smart home applications.

Building upon these insights, we provide a set of guidelines that emerge from a literature review and from an analysis of current interfaces and tools for end-user programming in smart homes. Starting from these guidelines, we present the design and the initial evaluation of HomeRules, a mobile and tangible application to empower home inhabitants with few or no technical skills to effectively "program" their smart home.

Literature Review

Several interfaces and tools for end-user programming in smart homes are present in the literature. Such applications, typically, let users define some *rules*[1] for programming, without writing any code, the environment where they live or operate.

To extrapolate a set of guidelines for designers of future end-users programming tools for smart homes, we first assembled a list of works known to the authors. We then completed the list by adding any of the search results on ACM and IEEE Xplore Digital Libraries for the keywords "end-user programming", "trigger-action programming", "activity delegation" and "context-aware computing" combined with "smart home" or "home automation", and not already included in the known body of works. The final list comprises 47 papers, counted after cleaning the results from unrelated works and excluding patents, keynote talks, front covers and similar content. In this paper, for brevity, we report a few of the most relevant works, only.

Among the analyzed papers, the work of Dey et al. [2] is, probably, the most cited and can be considered as a "cornerstone" for several subsequent papers. In their work, they present iCAP, a visual, PC-based and rule-based system for building context-aware applications for smart environments that does not require users to write any code. iCAP supports trigger-action rules ("if

[1] also called "context-aware applications" or "trigger-action programs"

something happens, then *do something else*") with "and" and "or" conditional operators. The trigger-action format is one of the most used in the analyzed papers and can be useful and usable for end-user programming in smart homes, according to Ur et al. [7]. Trigger-action is sometimes extended to trigger-constraint-action, also known as Event-Condition-Action (ECA).

Lee et al., in [3], propose GALLAG Strip, a mobile and tangible tool for creating context-aware applications. Various insights emerge from the in-depth validation of GALLAG Strip, like the suggestion of mixing tangible and non-tangible approaches for augmenting the overall user experience or the limitation of using the "small" screen of a smartphone. Other interesting suggestions are reported in the paper of Ur et al. [7], where the authors investigate the practicality of end-user programming to customize smart home devices, evaluating thousands of trigger-action programs shared on the website IFTTT ("If This Then That")[2] and conducting a usability test with more than 200 participants.

Guidelines

The guidelines proposed in this section synthesize the findings that emerge from the literature review and aim at providing a clear guidance for designers and engineers interested in the creation of end-user programming tools for a smart home. They avoid giving any insight about rule clashing, conflicts or debugging. In writing such guidelines, we also consider our own previous experiences and some well-known commercial products that enable average users to engage in trigger-action programming with household devices, like IFTTT or the WigWag system[3]. The extracted guidelines are:

1. *Use the ECA format for representing smart home rules, in a visual way.* Trigger-action is one of the most common formats in the reviewed papers, a good match with users mental model [2] and can be useful and usable for end-user programming in smart homes [7].

2. *Avoid using natural language for rules creation.* Typically used words for specifying events and conditions (i.e., "if" and "when") are semantically close in many languages. This may confuse the users during the rule creation process.

3. *Provide a simple and clear visualization of existing rules.* With a simple and clear visualization of rules, home inhabitants accept and use better their smart home, especially those who have a lack of interest in participating actively in the configuration of smart home technologies.

4. *Provide both tangible and non-tangible interaction.* A mixed tangible and non-tangible approach seems appropriate to encompass users with different needs and programming skills [3].

5. *Provide a step-by-step creation mechanism for first-time users.* Novice users can largely benefit from a step-by-step rule creation mechanism, to better understand the trigger-action model and the meaning of "creating a smart home rule".

6. *Handle time-related properties separately.* Temporal relationships inside rules (e.g., "at 6 o' clock", "for 2 minutes", etc.) and chronological operations should be treated in a dedicated, specific way to isolate their complexity.

[2] http://ifttt.com, last visited on December 29, 2014
[3] http://www.wigwag.com, last visited on December 29, 2014

7. *Apply a mobile approach, targeting screens larger than 6".* Small screens can be a issue during rule creation [3].

8. *Adopt single touch (or click) interactions instead of more complex operations.* As reported in the following section, users seem to find "easier" the usage of single touch (or click) for rule composition instead of more complex operations, like drag-and-drop.

9. *Supplement the visual representation with other feedback mechanisms, like sound.* Particularly useful for tangible programming, sound or other type of feedback can signal the success of an operation without requiring the user to look at the screen.

10. *Prefer testing "in the wild".* Several issues that can emerge in realizing a rule composition tool are difficult to catch and understand in a controlled setting. Evaluation should, therefore, be performed by including also sessions in the field, given the great availability of commercial smart home devices.

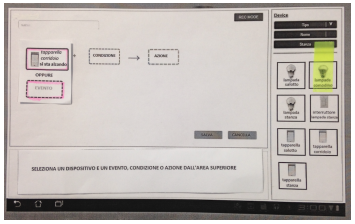


Figure 1: The "single touch" paper prototype.

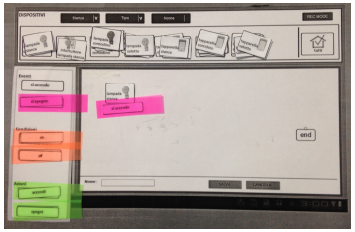


Figure 2: The "drag-and-drop" paper prototype.

Initial Prototypes

With the previous guidelines in mind, we designed two paper prototypes with the goal of creating a tablet-based, end-user and tangible programming interface for a smart home.

Both prototypes express a rule in term of events, conditions and actions (according to the ECA model); in particular, each rule is triggered by an event only (or a series of mutually disjoint events), have to optionally satisfy a set of conditions, and may execute one or more actions. The prototypes present three areas: a *Devices Area* that collects all smart home devices; a *Composition Area*, used for the actual rule composition; and a *Choice Area* that shows the events, conditions or actions available for the selected device. The two prototypes mainly differ for the modality in which a rule can be composed: the former (Figure 1) uses single touch for rule composition, while the latter (Figure 2) employs drag-and-drop for the same operation.

Three HCI/Ubicomp experts evaluated the prototypes to identify possible weaknesses in the concepts and the general interface design. After a minor revision, those prototypes were evaluated using a think-aloud, within-subject study with two groups of 6 participants (3 male, 3 female) each: one group composed by people with little to no technical experience and no programming experience, and the other composed by participants with a computer science or engineering background. Participants ages ranged between 24 and 30, with a mean age of 26.2 years (SD = 2.6) for the "non-technical" group and of 25.7 years (SD = 2.3) for the other. All of them lived with at least one more person and without their parents. Our main goal in this initial evaluation was to learn about whether participants understood the presented ECA format, which interaction modality they preferred, and whether they envisioned using them.

Participants in both groups composed successfully at least one rule with each prototype and indicated that they understood the used format and formalism. Both groups strongly preferred the interaction modality of the first prototype (i.e., single touch) as they found it "*simpler*" and "*less demanding [from a cognitive standpoint]*". Two non-technical participants had some minor difficulties in understanding the difference between an event and a condition: they suggested to better highlight this difference, maybe using different icons for distinguishing

the events that trigger the rule from the conditions. When asked about whether they will use a tangible approach for composing such rules, half of the participants (3 for each group) found it useful but not essential, thus confirming the fourth guideline. Because our participants expressed a strong interest in the first prototype, we decided to focus our subsequent efforts on it, developing an interactive prototype named HomeRules.

Interactive Prototype: HomeRules

To better investigate the applicability and the consistency of the reported guidelines, we revised the HomeRules design, addressing issues identified in the paper prototype evaluation and adding missing functions. The paper prototypes evaluated by the end-users, in fact, cover guidelines 1, 2, 3, 4, 7, and 8, partially or completely. The interactive prototype completes the tangible part (guideline 4), adds the step-by-step rule composition mechanism (guideline 5) and handles temporal relationship (guideline 6). It also adds the various menus and feedback mechanisms (guideline 9). All these functions were evaluated by the three experts.

In particular, the tangible interaction is enabled from the rule composition screen by choosing the "*Interactive Learning*" option. This modality puts the application in a listening mode: HomeRules waits until the user performs any operation in the home and then reports it. In case of ambiguity, the application lets the user choose whether the perceived operation is an event, a condition or an action (Figure 3). The step-by-step rule composition is realized in two modalities: a "traditional" guided tutorial and a "*suggestion*" modality that shows the next operation(s) to be performed to successfully compose a rule (like in Figure 4). Up to now, temporal relationships are handled through a "clock" and a "calendar" virtual devices, which can be used as events or conditions.

The current prototype of HomeRules is an Android 4.x app (see Figure 5), optimized for a 10" screen. HomeRules takes the needed data from an instance of Dog[4], an open-source smart home gateway, that runs in one of our University offices. HomeRules uses the Dog WebSocket API for getting the list of the smart home devices, their properties, and for interacting with the devices in the tangible modality. In the current state, all the rules created with HomeRules are saved in JSON format and sent to the Dog gateway, that is responsible for their execution. In the same way, at start time, HomeRules asks Dog for already existing rules to allow their editing.

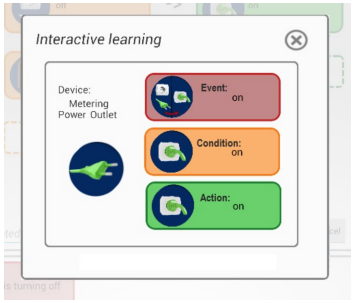


Figure 3: The "Interactive Learning" modality at work.

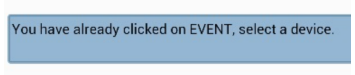


Figure 4: A possible suggestion as shown in HomeRules.

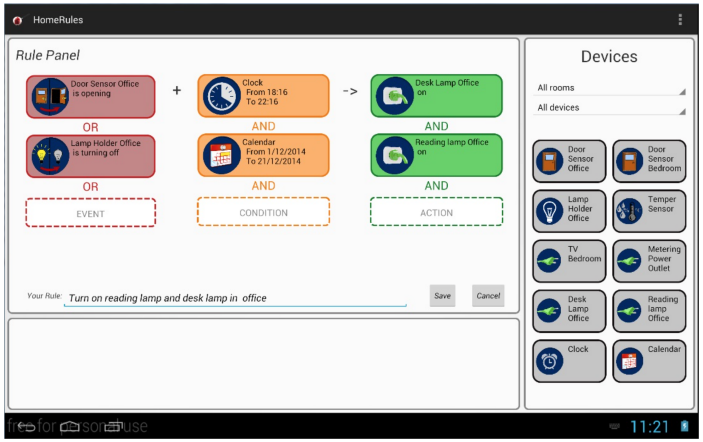


Figure 5: Screenshot of the current version of HomeRules.

[4] http://dog-gateway.github.io, last visited on December 29, 2014

Conclusions and Future Works

The contribution of this work in progress is twofold. From one side, it proposes a set of guidelines, extrapolated from the literature, that should constitute a strong foundation and starting point for designers and engineers of future end-user programming tools for smart homes. From the other side, it presents HomeRules, a prototype interface that builds upon the reported guidelines and seems to confirm their validity; moreover, its preliminary evaluation better clarifies and details most of the guidelines.

Currently, we are preparing a user evaluation of the interactive prototype of HomeRules. As a future work, we plan to extend HomeRules to encompass issues like visualization and explanation of rule conflicts, debugging and preview of created rules, and high-level handling and concatenation of multiple rules. Finally, we aim at developing a prototype suitable for real-world deployment, to better understand the effective usage of the application and obtain further information on how end-user programming can improve the smart home experience.

Acknowledgements

The authors wish to thank the MSc student Rosalba Castro for developing the interactive prototype and all the anonymous participants of our study.

References

[1] Ball, M., and Callaghan, V. Managing control, convenience and autonomy - A study of agent autonomy in intelligent environments. In *Agents and Ambient Intelligence - Achievements and Challenges in the Intersection of Agent Technology and Ambient Intelligence*. 2012, 159–196.

[2] Dey, A. K., Sohn, T., Streng, S., and Kodama, J. iCAP: Interactive Prototyping of Context-aware Applications. In *Proceedings of the 4th International Conference on Pervasive Computing*, PERVASIVE'06, Springer-Verlag (Berlin, Heidelberg, 2006), 254–271.

[3] Lee, J., Garduño, L., Walker, E., and Burleson, W. A Tangible Programming Tool for Creation of Context-aware Applications. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, ACM (New York, NY, USA, 2013), 391–400.

[4] Mennicken, S., and Huang, E. M. Hacking the Natural Habitat: An In-the-wild Study of Smart Homes, Their Development, and the People Who Live in Them. In *Proceedings of the 10th International Conference on Pervasive Computing*, Pervasive'12, Springer-Verlag (Berlin, Heidelberg, 2012), 143–160.

[5] Mennicken, S., Vermeulen, J., and Huang, E. M. From Today's Augmented Houses to Tomorrow's Smart Homes: New Directions for Home Automation Research. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, ACM (New York, NY, USA, 2014), 105–115.

[6] Newman, M. W. Now we're cooking: Recipes for end-user service composition in the digital home. In *CHI 2006 Workshop IT@Home* (2006).

[7] Ur, B., McManus, E., Pak Yong Ho, M., and Littman, M. L. Practical Trigger-action Programming in the Smart Home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, ACM (New York, NY, USA, 2014), 803–812.