Supporting Users in Controlling Automations in Intelligent Environments Through Conversational Agents

Simone Gallo [1,2]

[1] *HIIS Laboratory, ISTI-CNR, Pisa, Italy*
[2] *Department of Computer Science, University of Pisa, Pisa, Italy*

Abstract

The integration of sensors and intelligent devices in everyday environments is changing the way people interact with common objects. The need to provide non-expert users with an easy and efficient way to customize the behaviour of these devices according to their preferences and habits is crucial. This paper presents the work carried out by the author in the development of a conversational agent to empower the end-user in controlling intelligent environments, allowing for easy and fast communication with sensors, actuators, and smart objects using trigger-action customisation rules. The paper provides an overview of the state of the art in research and commercial solutions for conversational agents and presents some preliminary results of the research. Then, the ongoing work on a Systematic Literature Review and the migration to open-source solutions are presented. Finally, an overview of future work is given.

Keywords

Conversational Agents, End-User Development, Smart Environment

1. Introduction

The spread of sensors and intelligent devices of the Internet of Things and their integration into daily environments are changing the way we interact with some of the most common objects in everyday life. Therefore, there is an evident need to provide non-expert users with the ability to customize, in a simple but effective way, the behaviour of these devices based on their preferences and habits.

To this aim, some commercial and research solutions have been proposed, but only a few of them use natural language to allow communication between end-users and the various sensors and smart objects present in intelligent environments.

This work aims to propose the development of tools and techniques to empower end-users in controlling intelligent environments through conversational agents, trying to enable an easy and fast, yet powerful way to control and communicate with sensors, actuators, and smart objects in daily environments such as smart homes.

The following pages describe the state of the art that presents some of the most interesting research and commercial solutions that move in this direction. Then, some results already accomplished are presented, followed by ongoing work and plans for future research.

2. Related Work

Several tools have been developed for creating trigger-action routines, both for commercial purposes (e.g., Amazon Alexa) and for research. Among these different solutions, most of them use a "classic" graphical interface, e.g., they make use of buttons and text fields for the creation of rules. Other tools, developed in the research field, make use of graphic interfaces, or explore textual and/or speech

IS-EUD 2023: 9th International Symposium on End-User Development, 6-8 June 2023, Cagliari, Italy
EMAIL: simone.gallo@isti.cnr.it (S. Gallo)
ORCID: 0000-0002-5162-0475 (S. Gallo)

methodologies. In the next sections, the most relevant works are presented, highlighting points of strength and weakness of each tool.

2.1. Commercial Solutions

Among the now widely used applications for controlling IoT devices within the home environment is Amazon Alexa. In addition to the "classic" functions related to performing instantaneous actions such as turning on the lights or music, there is the possibility of creating routines using the graphical interface provided by the corresponding smartphone application.

This allows users to add a single trigger (called "When" or "When this happens") and an indefinite number of actions. The former can refer to the devices installed inside the home and compatible with Alexa, upon the occurrence of conditions such as sentences spoken by the user or the occurrence of a certain time. The actions can instead activate functionalities related to the launch of services (such as weather, music, news) or control devices in the home (e.g., turning on a lamp).

Another of the most popular commercial solutions is Google Assistant. The functionalities available are almost the same as those offered by Alexa, giving the user the possibility to execute an action (such as adjust home devices, send reminder or play and control media) when a condition is satisfied (e.g., when the user says an activation phrase, at a certain time or when a device changes its state).

Less well known than the ones just mentioned, but still quite widespread is IFTTT. This is an online service that offers the possibility of creating rules in a trigger-action format (called applets) that give rise to simple applications for controlling smart objects or web applications.

It is possible to create triggers or actions covering a wide range of services, such as social networks, weather, Google services, receiving calls and instant messaging services. It is also possible to integrate available actions with routines created via Alexa.

IFTTT allows the creation of rules consisting of a single trigger and a single action using the free version, while you can use a potentially unlimited number of triggers and actions using the paid version.

2.2. Research Solutions

The services discussed so far share the possibility of creating automations using classic graphic interfaces, and the impossibility (despite the nature of the devices with which they are associated) of realising them using a vocal interface, or, more generally, based on the use of natural language.

The following are some applications that make use of textual or multimodal interfaces (both vocal and textual) for the creation of trigger-action rules. The solutions presented are not commercial applications, but the result of research work. At the time I am writing this proposal, there are no tools on the market that enable the processing of such rules using spoken or written language.

HeyTAP is a conversational agent developed at the Politecnico di Torino for personalising the behaviour of smart objects in a home [1]. It presents a multimodal interface through which the user can express preferences regarding the behaviour off installed devices. After expressing its intentions, the chatbot proposes to the user a set of predefined rules extracted from IFTTT that 'come as close' as possible to what was requested by the user, based on the available devices.

In this case, the limitation of HeyTAP is that it requires the trigger part of the rule to be defined individually and separately from the action to be performed. Furthermore, it does not allow the direct creation of the rule but requires further interaction with a "classic" interface in order to choose the routine from those proposed by the system, that comes closest to the user's request.

Another application that exploits natural language for the creation of trigger-action rules is **Jarvis**. Developed at the University of Porto, it was implemented using Google's Dialogflow framework. Jarvis makes use of a textual interface – accessible via the instant messaging application Slack – and a vocal interface thanks to the integration with Google Assistant [2].

The main objective of the Jarvis project concerns the spatial and temporal contextualisation of the behaviour of IoT devices, which is not present in the main virtual assistants on the market. The main functionalities allow the creation of such rules or the instantaneous or delayed execution (One-time action and Delayed Action) of commands on devices inside the home (think Alexa or Google Home).

Inputs relating to the creation of customisation rules allow for a maximum of one trigger and one action, so possible inputs could be “Turn on the light when the bedroom sensor is triggered”.

The execution of delayed actions, on the other hand, requires the declaration of an action to be executed at a certain time and day of the week, e.g., “Turn on the light tomorrow at 5 p.m.”.

Although not strictly related to the Internet of Things, it is worth mentioning **SUGILITE** [3], a system that uses the programming by demonstration (or programming by example) paradigm for the creation of automations on mobile devices. Implemented in Java with the help of the Android API, this software makes its functionalities available via an app for mobile devices. The user can then associate a set of actions with a voice command, showing – via direct navigation through the applications installed on the device – how these are to be performed. The system, after recording the steps required to achieve the goal, creates a script that will be executed whenever the user so wishes, giving the more experienced also the possibility of making changes directly from a code representation.

For example, the command “order a cappuccino” could be associated with the actions of opening the ordering application, selecting the café from which to order, and selecting the desired product. In addition, the generated script has a certain degree of generalisation that allows the variation of certain parameters of the request without the need to show the process to be performed again. Taking the example just given, it would be possible to change the product to be ordered since, in the command “order a [cappuccino]”, the term “cappuccino” is a parameter that can be replaced with any other product on the menu of the bar in question.

A further contribution is given by Barricelli et al**. [**4] with a multi-modal approach for creating smart speaker routines through an Alexa skill, giving to the user the possibility to create routine for Alexa using both voice and touch on Echo Show devices (the ones with a display). When the user says the skill activation phrase (“Alexa, start create a new routine”), the virtual assistant engages a conversation structured along three steps: first asks for the routine name, then guides the user in choosing the trigger and finally asks the user to select one or more action. If necessary, the virtual assistant will ask for the necessary information to configure triggers or actions (e.g., in the case of an action on lights, the assistant will ask for the room in which the light is to be switched on if not specified). These interactions can be carried out through voice commands or by selecting relevant options on the device's touchscreen.

Valtolina et al. [5] reported on a study evaluating the benefits of a chatbot in comparison to traditional GUI, specifically for users with poor aptitude in using technologies. They considered two example scenarios in the healthcare and smart home fields and found that for the user experience the chatbot application appears to be better than the GUI-based one.

Despite such promising indications, the conversational interaction style has received limited attention for supporting the creation of trigger-action rules in general. This proposal, with the research done during this year, and that which will be done in the following years, aims to contribute to cover such gap.

3. Preliminary Research Results

In this section, the work carried out during the first year of the PhD is presented. First, the research done to develop a conversational agent for personalising sensors and smart objects in home environments is summarised. Then, the ongoing work on a systematic literature review to study how conversational agents are used to control and personalise smart environments is briefly presented.

3.1. A Conversational Agent for Creating Flexible Daily Automations

This work [6] presents the design, development, and testing of a conversational agent able to support the end-user in the creation of customisation rules for smart home environments, using the trigger-action programming paradigm.

The trigger-action programming language we adopted can clearly distinguish between events and conditions, can use multiple triggers composed using logic operator AND and OR, while multiple actions are executed sequentially.

In particular, the language is organised based on three main dimensions:

- User: includes aspects related to user physical activity, physical status (e.g., heart rate, hours of sleeping) and emotional status.
- Environment: includes aspects related to the surrounding elements such as light, temperature, noise, humidity, and related services like weather forecasts.
- Technology: includes aspects related to smart devices such as the state of PCs, smartphones, TVs and so on.

When events and/or conditions are satisfied, is possible to activate actions that can change the state of smart devices (e.g., turning on/off the lights), activate device-based service (e.g., Alexa) or send reminders and alarms.

For the development of the conversational agent, we followed an iterative design process: three different versions were developed and tested to identify weaknesses and strengths.

3.1.1. First and Second Versions

The first version (V.1) was able to create the rules "piece by piece", which means that the user was only able to insert input that can include one trigger or one action at a time. This type of interaction was dictated by the way of implementing this version: in fact, to achieve great accuracy in the input recognition, we created a single intent for each trigger and action present in the language (a total of 66 intents were created for both triggers and actions). Then, for each intent, the set of training phrases includes sentences that referred directly to the single element considered. For example, the intent for the recognition of inputs related to the smoke sensor was trained with sentences like "If the smoke sensor in the kitchen is active", or "if is activated the smoke sensor in the living room".

Moreover, the training phrases used to train the model had a "device-centric" level of abstraction [7] (e.g., "when the motion sensor in the kitchen becomes active"), for which the user inputs must contain a direct reference to the sensor to be used, forcing the user to conform to this specific way of looking at automations.

This version was tested by twenty-three students of Digital Humanities that described the system as somewhat imprecise, requiring several interactions, and thus not very efficient. The conversation was judged fragmented and unnatural, especially for rules that include more than one trigger and action.

The second developed version (V.2) aims to the possibility of receiving multiple triggers and actions within a single user input.

For example, in the input sentence "when I enter the living room, turn on the lights" we identify a trigger ("when I enter the living room") and an action ("turn on the lights"). For the intent structure implemented in V.1, it is impossible to categorise a single input to several triggers or actions requests, because these have been defined as single intents.

This second version presents an intent structure to categorise training phrases that include, for example, one trigger and one action, or two triggers and two actions. Initially, three main intents were implemented, that considers input containing:

- two triggers: trained with phrases like "if it's raining or snowing", or "when I enter the bedroom and it's dark";
- two actions: trained with phrases like "send me an alarm and turn on red lights at home", or "turn on the light and play some music";
- one trigger and one action: trained with phrases such as "turn off the lights when I leave home" or "send me a notification if it will rain tomorrow".

However, this approach raised several problems regarding the quantity and the quality of the training phrases; the recognition of the specific trigger and action present in one input and the scalability of the corresponding architecture. Thus, we moved to the following final solution.

3.1.2. Final Solution

We thus designed and developed a new solution (V.3) with the aim of overcoming the limitations of the previous ones. The new design implements three different components: a "**Dialogue Interface**"

used for receiving user's input and for sending chatbot responses, an "**Intent Classifier**", and a "**Dialogue Manager**" interposed between the two. The Dialogue Manager covers the central role in the management of the system. It handles conversational turns, processes user inputs, generates bot responses, and manages all the secondary functions such as saving or deleting rules.

The main functionality of this component concerns the processing of complex user inputs. In particular, when the user sends an input that describes a personalization rule containing multiple triggers and actions, the Dialogue Manager applies regular expressions and Part-of-Speech tagging to split the complex sentence into smaller pieces (based on some grammar rules). Then, the extracted sub-sentences that represent the single triggers and actions involved are sent, one by one, to the Intent Classifier to perform the intent classification and the extraction of parameters used to correctly configure the trigger and actions involved. These data are then sent back to the Dialogue Manager, and after a validation process, it generates the response for the user.

This final version was tested on 10 users. Each user was asked to complete four tasks of increasing difficulty, using two different tools: the chatbot and a platform that uses a classic graphical interface to create personalisation rules. Finally, each user filled out a questionnaire in which they were asked to express their opinions about the efficiency and their appreciation of the experience with the two tools.

To measure the chatbot performance during the task completion, we took into account different metrics: the time employed to create a rule, the number and severity of errors on the rule created (considering both the full rule and the errors in the single trigger or action involved), and the number of conversational turns to create a rule (from the initial input to the rule saving message).

The results have pointed out good efficiency and usability of the chatbot system, described as fast and easy to learn and use. Negative comments are, instead, related to the accuracy of the system. In some cases, the wrong results of the "splitting algorithm" or a wrong intent classification led to the creation of incorrect rules.

The implementation of these prototypes was done using the Dialogflow framework associated with a Node.js server and some Python scripts to perform further text analysis.

4. Ongoing and Future Work

At the moment of writing, we are putting forward two main works: a Systematic Literature Review on conversational agents for controlling and automating smart environments, and the migration from the Dialogflow environment to the Rasa Open Source framework, which is open-source and allows developers to better control the dialogue management, choose the machine learning algorithm to use to classify users' intentions and define custom natural language processing pipelines.

The Systematic Literature Review aims to explore and analyse the various work in the field of conversational agents applied in interaction with fully intelligent environments, or with a set of smart objects. The study wants to address the following research questions:

- Q1: What technologies are used to build a conversational agent? (e.g., rule-based, ML/DP, Framework, Web Platform)
- Q2: In what Internet of Things application domain are conversational agents used?
- Q3: What evaluation methodologies are used to measure usability, accessibility, and user experience?
- Q4: How much AI is involved in conversational systems? (e.g., NLP, recommender system, smart IoT devices configuration)

The research has been conducted in the digital libraries of ACM and IEEE in November 2022, retrieving 2498 articles from the digital libraries (1805 from ACM, 693 from IEEE). Other 284 articles were retrieved from the journals and conference proceedings (17 from IS-EUD, 249 from Multimedia Tools and Applications and 18 from Behaviour & Information Technology).

Then, after removing duplicates papers and works not in line with our interest, we applied the following selection criteria to 79 remaining articles:

- Papers must be written in English;
- Papers must address the application of dialogue system to IoT environment, excluding work with single device;

- The dialog system must use natural language and use a conversational approach. We don't consider application that uses predefined commands (such as some telegram chatbot that uses commands like "/start", "/createItem", "/stop" to communicate);
- Papers length must be equal or major of 4 pages.

Finally, we have a set of 49 papers that passed the selection criteria. Given the spread of Large Language Models (LLMs) and the consequent evolution of conversational agents, we are now working on additional research starting from papers published after November 2022.

Regarding the migration to Rasa Open Source, the overall idea is to overcome the limitations of the current solution based on Dialogflow (e.g., errors in interpreting complex rules, limited capacity to manage breakdowns), improve its capabilities and move to open-source. To achieve these improvements, we plan to build a hybrid solution in which Rasa functionalities are augmented by using an LLM to solve different natural language processing tasks.

Moreover, based on the work done so far, future work will focus on two different but complementary branches of conversational agents. The first is strictly related to transparency and control through conversational agents in smart homes (or more generally intelligent environments).

The objective aims to improve the capabilities of the conversational agent filling some of the state-of-the-art gaps by providing the users with the ability to organically address various needs related to smart-home automation control, such as giving explanations about the environment and the automations, managing possible conflicts between automations, and acquire sensors data to suggest automations based on users' habits.

The second research line aims to manage conversational breakdowns and how the users perceive and manage the interaction with conversational agents, thus we want to explore how the users approach the conversation with a virtual assistant, focusing on error mitigation strategies and how to automatically learn from mistakes made.

Finally, we plan to conduct user tests in environments (such as a home) equipped with sensors, actuators, and smart objects to validate the effective usability and experience in real smart homes and compare this solution with other possibilities.

5. Conclusions

This paper presents the work carried out during the first year and half of the author's PhD and lays the foundation for future research in the next year.

In particular, this work regards the development of methods and tools that allow end-users without particular technological skills, to easily understand, control and personalise intelligent environments, such as smart homes, through the use of conversational agent.

To this purpose, in the first section the main research and commercial solution about conversational agents and smart environment are shown, highlighting its weaknesses and strengths.

Follow a description of the work done so far introducing RuleBot, presented at AVI 2022. Finally, an overview of the ongoing work (a systematic literature review and the migration of RuleBot to open-source framework) is presented, also giving the future direction of this project.

6. Acknowledgements

This Ph.D. project is supervised by professor Fabio Paternò and professor Alessio Malizia; supported by ISTI-CNR and University of Pisa.

7. References

[1] F. Corno, L. De Russis, e A. M. Roffarello, «HeyTAP: Bridging the Gaps Between Users' Needs and Technology in IF-THEN Rules via Conversation», in *Proceedings of the International Conference on Advanced Visual Interfaces*, New York, NY, USA: Association for Computing

Machinery, 2020. [Online]. Disponibile su: https://doi.org/10.1145/3399715.3399905

[2] A. S. Lago, J. P. Dias, e H. S. Ferreira, «Managing non-trivial internet-of-things systems with conversational assistants: A prototype and a feasibility experiment», *Journal of Computational Science*, vol. 51, p. 101324, apr. 2021, doi: 10.1016/j.jocs.2021.101324.

[3] T. J.-J. Li, A. Azaria, e B. A. Myers, «SUGILITE: Creating Multimodal Smartphone Automation by Demonstration», in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver Colorado USA: ACM, mag. 2017, pp. 6038–6049. doi: 10.1145/3025453.3025483.

[4] B. R. Barricelli, D. Fogli, L. Iemmolo, e A. Locoro, «A Multi-Modal Approach to Creating Routines for Smart Speakers», in *Proceedings of the 2022 International Conference on Advanced Visual Interfaces*, in AVI 2022. New York, NY, USA: Association for Computing Machinery, 2022. doi: 10.1145/3531073.3531168.

[5] S. Valtolina, B. R. Barricelli, e S. Di Gaetano, «Communicability of traditional interfaces VS chatbots in healthcare and smart home domains», *Behaviour & Information Technology*, vol. 39, fasc. 1, pp. 108–132, gen. 2020, doi: 10.1080/0144929X.2019.1637025.

[6] S. Gallo e F. Paterno, «A Conversational Agent for Creating Flexible Daily Automation», in *Proceedings of the 2022 International Conference on Advanced Visual Interfaces*, in AVI 2022. New York, NY, USA: Association for Computing Machinery, 2022. doi: 10.1145/3531073.3531090.

[7] F. Corno, L. De Russis, e A. Monge Roffarello, «Devices, Information, and People: Abstracting the Internet of Things for End-User Personalization», in *End-User Development*, D. Fogli, D. Tetteroo, B. R. Barricelli, S. Borsci, P. Markopoulos, e G. A. Papadopoulos, A c. di, in Lecture Notes in Computer Science, vol. 12724. Cham: Springer International Publishing, 2021, pp. 71–86. doi: 10.1007/978-3-030-79840-6_5.