Integrating Alexa in a Rule-based Personalization Platform

Marco Manca, Parvaneh Parvin, Fabio Paternò, Carmen Santoro
marco.manca,parvaneh.parvin,fabio.paterno,carmen.santoro@isti.cnr.it
CNR-ISTI, HIIS Laboratory
Pisa, Italy

ABSTRACT

Vocal assistants are becoming widely used, but their potentialities have not yet been completely exploited. For instance, while assistants such as Alexa are increasingly boasting compatibility with a large set of third-party services, the possibility for end-users to personalize the joint behaviour of such connected services (including the voice-based ones) in a flexible manner seems not sufficiently explored yet. In this paper, we present how the voice-based support offered by Alexa has been integrated with a rule-based personalization platform to support the creation of trigger-action rules enhanced with voice-based support. This integration opens up the possibility for users without programming knowledge to specify and include voice-based triggers and voice-based actions in their rules. These rules can be composed of events and commands that can involve a variety of sensors and connected objects. To this aim, a novel solution has been developed, which also aims to overcome some limitations that have been found in currently available vocal assistants, e.g., the issue of unsupported languages, thus lowering the barriers for their ultimate adoption and everyday use. Indeed, the integrated platform offers the possibility to play the vocal notifications/reminders contained in relevant personalization rules in any language, including those not currently supported by Alexa.

CCS CONCEPTS

• **Human-centered computing → User interface programming**; **Sound-based input / output**.

KEYWORDS

Ambient-assisted living, Smart-home, Voice assistant, Rule-based penalization platform

ACM Reference Format:
Marco Manca, Parvaneh Parvin, Fabio Paternò, Carmen Santoro. 2018. Integrating Alexa in a Rule-based Personalization Platform. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Voice-based personal assistants such as Amazon Alexa, Google Assistant or Apple Siri, have become highly popular nowadays [9], and it is predicted that their adoption will continue to increase in the near future [11]. Generally, they are embedded in smartphones or smart speakers and are used for a plethora of tasks as they provide people with a natural connection to their online searches, music, IoT smart objects, alarms, and wakes [2]. One aspect that makes voice assistants particularly interesting is the possibility of combining them with various external services, called 'Skills' in Alexa, or 'Actions' in Google Assistant, which are programmatic extensions that provide additional functionality toward a variety of actions, ranging from providing weather information, sending routine reminders and notifications, to controlling appliances and smart objects. As such, vocal assistants can be helpful to more easily support everyday life, also for people who live with a disability, illness, or ageing [10], in that they can help users in better managing their day-life routines (e.g., pill management) [5, 6].

One of the reasons for such potential is the effectiveness of the vocal modality compared to other kinds of interaction, which has been emphasized in various contexts in previous work. For instance, a systematic review by Robotham et al. [12] to assess the impact of text-based electronic notifications on improving clinic attendance, highlights that voice notifications appeared more effective than textual ones. Other studies highlighted the importance of analysing the use of vocal assistants by more fragile people such as older adults. In particular, in [15], the authors found the reasons why voice assistants have been used in a limited way, or sometimes even abandoned by seniors. They indicated that these issues are related to finding valuable use cases, questioning their independence (e.g. some older adults indicated that they still have the ability to complete tasks on their own, and they enjoy this freedom), and using vocal assistants in shared spaces. However, in order to be effective for the diverse population and settings they aim to address, vocal notifications need to be tailored to the user situation, current needs, and they need to be suitably conveyed, at the most proper time and using the most suitable media. More in general, with different users accessing various services, technologies and applications in diverse scenarios, there is a need for users to be able to directly shape and tailor their applications according to their current needs. End-User Development (EUD) [7] aims to provide users with usable means for personalizing their applications, and EUD trigger-action -based environments [3, 8] have been identified as particularly promising for their intuitiveness when creating automation involving connected objects in our daily life [16]. In these approaches, each rule is composed of trigger/s and action/s. A 'trigger' could be an event associated with a state change of a connected object, or a condition lasting some time (such as a specific interval of time, or a particular user state). Actions represent the effects that should be activated when a trigger is verified, and they can range from notifications, alarms, to controlling the state of devices and appliances.

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

Thus, as vocal assistants are becoming widely present in our homes, the opportunities offered by the integration of vocal personal assistants (such as Alexa) with a platform providing programmable and personalized support seemed worth investigating. This integration can offer possibilities such as voice-based triggers (e.g., based on users' vocal utterances, e.g. providing information on their emotional status), voice-based actions (e.g., vocal reminders), or controlling home appliances. However, when analysing more in detail the opportunities of integrating such vocal assistants with a programmable rule-based platform, several issues have arisen, hampering the possibility of offering voice-based enhancements within the personalized support. Indeed, vocal assistants, despite their broad adoption, are not free from limitations. One drawback is represented by the set of supported languages, which is limited to a narrow number of spoken languages recognised, thus excluding specific portions of the world population. Indeed, according to [14], Alexa is currently available in 7 languages and Google assistant in 13 languages. In order to cope with such limitation, Amazon introduced a new language-learning skill called Cleo [1] to teach Alexa new languages. However, it takes a long time to train Alexa in this way to be able to learn a foreign language; in addition, this skill is available for just Echo (Amazon smart speaker) owners who speak English and are also fluent in a second language. Another limitation is that not all smart devices are compatible with these vocal assistants. For example, Alexa mainly supports scenes with lighting, switch, and thermostat devices.

In this paper, we present how the possibilities offered by the Amazon Alexa, one of the most popular vocal assistants, can be integrated with a rule-based personalization platform to create rules able to execute actions (e.g., sending alarms/reminders, controlling appliances and devices) without requiring knowledge of the implementation languages used for programming Alexa or the connected devices and objects. This has been done by developing a novel solution to overcome some limitations found in current vocal assistants and to lower the barriers to their adoption and everyday use. The integrated platform also offers the possibility to play the notifications/reminders in any language, including those not currently supported by Alexa.

In the paper, after introducing a fictional scenario (Section 2) showing the possibilities of the integrated personalization platform, and the context in which the work has been carried out (Section 3), we present the architecture of the integrated platform in Section 4 and provide a description of the implementation in Section 5. Finally, some concluding remarks and suggestions for future work are presented in Section 6.

2 SCENARIO

Mihai is a Romanian heart specialist who lives in Austria. He has a personalized system in his house that allows setting-up various automation used by him and his family. For instance, he has set up a rule to activate a relaxing scene exploiting a luminaire in the living room when it is about the time to go to bed, and, a rule to remember him about his son's football matches schedule at school (because it is important for him to support his son at the play-field). Furthermore, he has an Echo Dot Alexa smart speaker, which has been integrated into the personalization system in order to open up many novel opportunities. For example, the system provides him with the possibility to reveal his emotional status (e.g., "I'm bored"), to activate actions such as inviting him to start an activity (e.g., "let's play your favourite game!"), or automatically carrying out some actions (e.g., turn on a specific light colour) to improve their mood.

Mihai has a father, Adrian, who is a 75-year-old Romanian man who lives alone in a flat in a big city in Romania. He suffers from diabetes, which he has to manage it by following a proper diet, doing regular physical exercises, regularly monitoring the blood sugar level, as well as taking some pills. Besides, he suffers from some mild episodes of memory losses and temporal disorientation which occurred especially after he retired, when his daily routine started to be somewhat disrupted. Despite this, Adrian is quite autonomous and active in his daily activities, although occasionally he feels a bit bored and isolated. Mihai thinks that this system could be very useful for his father also to better organize his daily routine: he would easily create some personalized automation rules for Adrian since he knows well his father's activities and problems. Thus, he started to investigate the possibility to use the personalization platform to remotely support his father. However, at some point, he found out that Alexa does not support the Romanian language: he was very disappointed for this since he thought that vocal communication can be immediate and effective in communicating messages to his father when some relevant event occurs (e.g., he forgets to do something, or he should be notified that there is some potentially dangerous situation).

However, after a while, he finds out that a new version of the personalization platform was able to render voice messages even for languages not supported by Alexa! This was great news because in this way he can provide his father with relevant notification/reminders in Romanian. This would be especially useful, because his father is also starting to experience some vision loss due to diabetes. Thus, he decided deploying the personalization platform in his father's home and started to create various personalization rules for him by programming personalized reminders rendered through Alexa using a combination of different events and conditions. In this way, the system e.g. can remind Adrian about the medicine that he must take at appropriate time and conditions, but also properly support him in managing diet and physical activities according to relevant contextual factors (e.g. depending on Adrian's current levels of blood sugar and/or outside weather conditions, it can invite Adrian to have a walk outside), by providing suitable messages in Romanian. Moreover, this system allows him to add a personalized message within the notification messages (e.g., it would be possible to add a description of the medicine box to take, to be able to more easily identify them, as well as specifying the dosage), and have them rendered vocally. When Adrian tried the system was so happy to have a tool that allows him to better manage his life, and to specify additional automation rules for obtaining other useful support in his daily life (e.g., reminders for medical appointments, for differentiated collection of waste, etc.). Thus, this system improves Adrian's self-care and supports a more independent lifestyle. More importantly, it enables Mihai to structure and support his father's schedules, tasks and reminders, when he cannot be there in person.

3 CONTEXT

This work has been carried out in the context of the AAL PETAL[1] (PErsonalizable assisTive Ambient monitoring and Lighting) project, which aims to increase elderly autonomy and assist them in carrying out activities of daily living. For this purpose, we developed the TAREME platform (Trigger-Action Rule Editing, Monitoring, Executing) [8] to enable people without programming knowledge (e.g., primary caregivers, but even elderly people having some familiarity with technology), to control the objects and devices in home settings. The platform can be applied to any application domain exploiting the Internet of Things (IoT) technologies such as smart retail, industry 4.0 and museums. The platform includes a visual Tailoring Environment that supports specifying personalization rules according to the trigger-action format, which has shown to be particularly suitable in IoT applications. The Tailoring Environment [4] helps to overcome issues existing in environments having similar goals, e.g., IFTTT, by clearly distinguishing events and conditions, and also supporting the combinations of multiple triggers and actions in the same rule (e.g., triggers can be combined through various Boolean operators like AND, OR, NOT).

Overall, the main platform components are *the Tailoring Environment, the Rule Manager and the Context Manager*. As mentioned before, the Tailoring Environment is a visual tool that allows end-users to define the triggers and the actions to include in rules. The tool is configurable for being used in different contexts/scenarios, and it offers triggers and actions that depend on the sensors, objects and devices currently available in a considered context. Triggers and actions have been categorised into two distinct hierarchies, which are visually rendered in the Tailoring Environment. The triggers' hierarchy is refined into three main contextual dimensions: users, environments, technology. The actions are grouped in terms of reminders, alarms, and functionalities associated with the available appliances. In case of reminders and alarms, it is possible to indicate a text, a notification mode (SMS, e-mail, push notification or vocal messages), a repetition (how many times the message should be repeated), and the target (e.g., an e-mail address or a mobile phone number). To specify rules, users navigate the hierarchy of triggers and the hierarchy of actions, by interactively selecting the concepts that are suitable for specifying the intended personalization rule. Users can save rules in a private or a shared repository. Depending on their evolving needs, they can select the rules they want to 'activate' for their possible execution, among those saved in the tool. When the user activates a rule for its execution, the Tailoring Environment sends it to the Rule Manager, which in turn asks the Context Manager to be notified when the corresponding triggers are verified. The Context Manager is composed of a Server and multiple Context Delegates that communicate with the associated sensors/objects to be informed when the state of such objects change: when this happens, they inform the Context Server, which updates a database accordingly. As soon as a rule is triggered, the Rule Manager sends (through an MQTT broker) the actions included in the rule to the relevant applications, i.e. those subscribed to the broker (under a specific topic) and which are able to interpret and apply such actions. Such a platform was applied in the first round of six trials carried out in three different countries (Italy, Romania, Austria) by using a set of off-the-shelf sensors, smart objects and devices. Example of sensors considered were motion, temperature, humidity, gas, proximity sensors. Smartwatches have been used to support indoor localization, exploiting Bluetooth communication for identifying the location (through proximity beacons), and WiFi communication to communicate to the platform the information on the current location.

[1]http://www.aal-petal.eu/

Various types of lamps have been supported to help older adults to better orient themselves temporally and spatially, and for providing activating and relaxing scenes. When we had to design the installation of the platform for the second round of trials, one of the new elements that were deemed interesting was the introduction of a vocal assistant for providing further support to older adults. The motivation was based on the fact that vocal assistants (with their associated speech recognition and synthesis services), can be potentially useful in providing support to older adults, and they seem a sufficiently mature technology to be introduced in household trials. Thus, it was decided to start with integrating Amazon Alexa in the platform. Such integration has given the possibility to extend the set of available triggers and actions. For the triggers, in the user-related contextual dimension, we added the features allowing users to vocally provide their current emotional state (i.e., happy, angry, fun, bored, serene, worried, enthusiast, sad), and to provide vocal commands (such as 'turn on a relaxing scene in the living room'). Regarding the actions, we have introduced support for activating the available Alexa services (e.g., news or weather forecast skill) when a rule is triggered, and the possibility to communicate reminders or alarms through the vocal channel. This makes it possible to execute rules such as "when the user has not taken a medicine between 8 and 10 in the morning, then the Alexa speaker should remind them by saying, "remember to take your pill, it is important for your health!".

As mentioned before, this work has been carried out within an international project in which, among others, trials have been planned also in Romania. Unfortunately, we found out that the Romanian language, among many others (e.g., Vietnamese, Dutch, Catalan, Russian), is not currently supported by Alexa, and neither by other voice assistants such as Google Home and it is unclear whether and when this support will be made available. Since we wanted to overcome this digital divide somehow, we started to investigate the possibility of finding solutions that, even if do not allow users to interact through such languages, they can allow rendering messages in unsupported languages and communicate notifications, reminders or alarms vocally through them.

4 THE INTEGRATION OF ALEXA WITH THE PERSONALIZATION PLATFORM

In this section, we describe how we have designed and integrated the support for Alexa in the TAREME personalization platform. The TAREME platform uses OpenHab (OH) [2] in its deployment, which is installed in gateways[3] placed in the homes where the trials should be carried out. OH is an open-source home automation software able to communicate with about 300 sensors/appliances manufactured by different producers; it has a modular architecture that can

[2]https://www.openhab.org/

[3]https://www.geniatech.com/product/gtw410/

be extended by developing new *add-ons* (an add-on is a software component that adds a specific feature to existing software) defined as an Open Services Gateway initiative (OSGi[4]) module. Within the personalization platform OH acts both as an 'application' (in that, it receives and applies the actions received by the Rule Manager), and as a 'context delegate' since it can receive information from several associated sensors, and updates accordingly the Context Server. The integration of Alexa in the personalization platform has a two-fold goal. On one side, Alexa vocal input can be used as a trigger, and on the other side, it can be seen as a target of an action, including playing vocal messages even for unsupported languages. Thus, we developed two different add-ons for OH as follows.

(1) the Context Delegate Add-On that gets the data from the connected sensors and appliances by exploiting an 'OH binding'. OpenHab bindings are software packages that integrate physical hardware, external systems and web services in the OH system. Thus, the Context Delegate Add-On has a handler that bounds to various sensors and appliances managed by OH: every time they change their state it gets informed and notifies the Context Server accordingly.
(2) the Rule Manager Add-On subscribes to an external MQTT broker for a specific topic, in order to receive the actions coming from the Rule Manager. When the Rule Manager add-on receives an action, it can apply it in different manners, depending on the type of the involved action (this will be better detailed later on in this section).

For introducing the possibility of receiving new triggers involving Alexa (i.e., the vocal emotional state and the user vocal input) we extended the Context Server in order to receive and manage them. By exploiting such triggers, users can set up a rule, providing their emotional state as a trigger (e.g., by saying "I am nervous"). This can be done by activating the Alexa skill developed for this aim, in which Alexa asks the user about his feeling and the user will provide the corresponding answer. Subsequently, a specific action can be activated, e.g., either inviting the user to start an activity (e.g., "let's go outside for a walk!"), or automatically changing the state of smart object installed in the house (e.g., "turn on a light with a specific activating colour"). Moreover, users can also indicate and execute rules that change the home appliances' status (even those not recognizable by Alexa) such as lights, colour and scenes through vocal inputs. In this case, the vocal inputs are gathered through a skill able to recognize the inputs and send them to the Context Server, which will notify the occurrence of the event and thus trigger the corresponding rule.

The integration of the platform with Alexa is described in Figure 1. In particular, as soon as events are generated by the various triggers contained in the user's home (1), they are sent to the Context Server (2), which stores them and updates accordingly its repository. When a trigger contained in a rule is verified, the Context Server notifies the Rule Manager (3), which then sends the actions to the OH Rule Manager add-on able to interpret and apply them (5).

If the action involves an appliance (e.g., a light), the Rule Manager Add-On can exploit the OH binding mentioned before to interact with such light. If the action specifies a reminder or an alarm that should be vocally rendered, it acts differently, depending on the language of the message: if it is supported by the considered vocal assistant (Alexa), it exploits the OH binding for Amazon Echo devices[5]. This binding is able to exploit the echo device as a Text-To-Speech service, by sending a request containing the text of the alarm (or the reminder) defined in the received action, so that the smart speaker can vocally render it (6a and 7). However, this approach works only for messages defined in languages currently supported by Alexa (e.g., Italian, German, English). If the message is written in a language not yet supported, it is not possible to exploit the Amazon Echo Binding, because the message will be synthesised in a wrong manner.

To overcome such limitation, we propose a solution that first gets the user-defined text (used for specifying the concerned personal notification, alarm, or reminder) from the personalization rule, then it generates the corresponding audio file and store it. This file will then be played through the vocal assistant, by exploiting SSML language (Speech Synthesis Markup Language) tags[6]. In this solution, we detect the language used in the alarm/reminder action by exploiting the Google Cloud Translation API[7]. If the language is not supported by Alexa, the Rule Manager converts the text of the message in an audio MP3 format, through an off-the-shelf text-to-speech synthesizer (provided by Google), and the resulting audio file will be stored in a server and will be made available through a REST service to be played by Alexa. Thus, when a rule concerning an unsupported language notification action will be triggered, it will be sent to the OH module. The OH module can start an Alexa automation routine (6a) whose purpose is to trigger the Alexa skill to play the audio file located in the concerned server (6b). More detail on the procedure of activation the Alexa skill through the OH is presented in Section5.1.

5 IMPLEMENTATION

In this section, we provide further technical details about the developed solution, in particular for generating vocal messages in languages not supported by Alexa.

5.0.1 Generation of Vocal Messages in Languages not Supported by Alexa. The generation of vocal messages in languages not supported by Alexa has been obtained by exploiting Text-To-Speech (TTS) technology. This technology was an important research area for many years, but it has known significant improvements in the last years, thanks to the application of deep neural network techniques [13]. In this regard, several solutions are available. Among all solutions, we used Google Cloud TTS [8], which provides a service that takes a text as input and outputs an MP3 file representing the synthesis of the provided text. Google Cloud TTS also offers an enterprise API for commercial scopes. However, for research purposes, we exploited the free online API provided by Google Translate Service. The service providing the audio file is composed of seven parameters: the character encoding (*ie*), the text to be synthesized (*q*), the target language (*tl*), the number of the messages that should be generated (*total*), the initial character index (*idx*),

[4]https://https://www.osgi.org/

[5]https://www.openhab.org/addons/bindings/amazonechocontrol/
[6]https://developer.amazon.com/it-IT/docs/alexa/custom-skills/speech-synthesis-markup-language-ssml-reference.html#audio
[7]https://cloud.google.com/translate/docs/basic/detecting-language
[8]https://cloud.google.com/text-to-speech

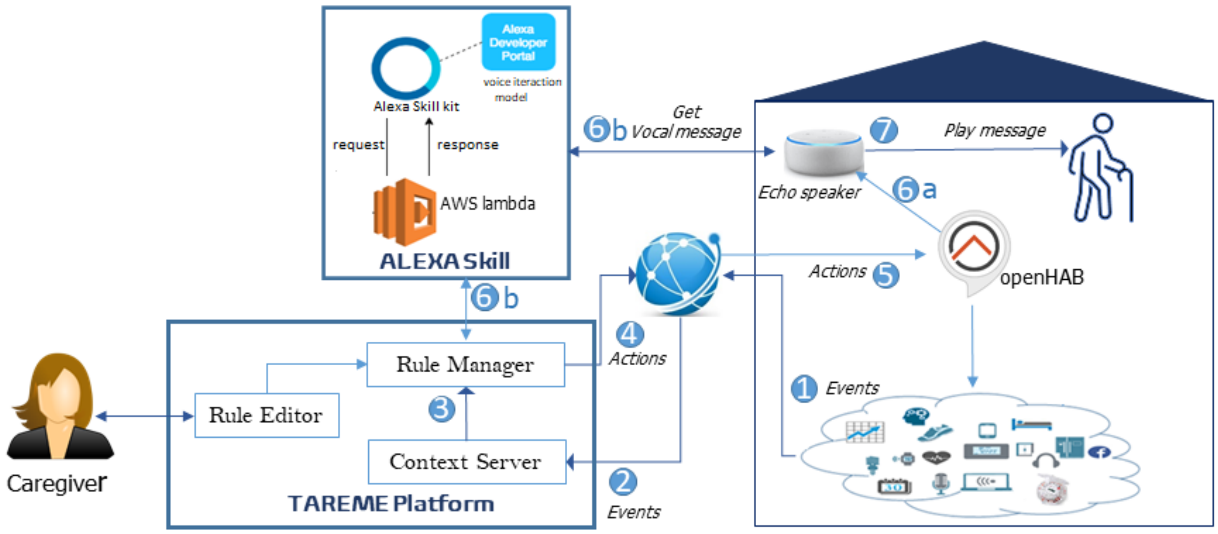

Figure 1: The personalization platform integrated with Alexa

the length of the text (*textlen*), a randomly generated number (*tk*), and the service used to call the TTS functionality ("*client*"). The *tk* parameter changes for every query, and it must be instantiated through a matching $hash : tk = hash(q, TKK)$, where q is the text to be synthesized and *TKK* is a variable in the global scope when the API will be loaded. Below, is the function to generate the google TTS link using the *calcHash* function.

```
function TTSLink(q, tl, tkk) {
  var tk = calcHash(q, tkk);
  return
   `https://translate.google.com/translate_tts?
   ie=UTF-8&total=1&idx=0&client=t&ttsspeed=1&tl
   =${tl}&tk=${tk}&q=${q}&textlen=${q.length}`;
}
```

The function call $TTSLink('salut!', 'ro', '410353.133636')$ is an example of generating the audio file for rendering in Romanian the 'salut!' text. Finally, in order to use the free TTS API provided by Google Translate Service, the following steps have been carried out:

(1) Get the *TTK* private key from the Google Translate website;
(2) Calculate the value of the *tk* parameter by applying the hash function which takes in input the text and the *TTK* parameters;
(3) Call the API with the parameters *ie*, *q*, *tl*, *textlen* and the calculated *tk* value;
(4) Store the received an audio file

As explained in the Alexa Skill Kit Developer documentation [9], in order to be rendered properly, the audio file must respect the following constraints: it cannot be longer than 240 seconds, the bit rate must be 48 kbps, and the sample rate must be 22050 Hz, 24000 Hz, or 16000 Hz. Since the file provided by the Google Translate service is an MP3 file with a 32 kbps bitrate, it cannot be played through the SSML audio tag. In order to make the generated file compliant with the needed requirements, the Rule Manager automatically modifies the bit rate of the downloaded file by using FFmpeg tool[10]. Later, the Rule Manager associates an ID to each personalized vocal message generated and exposes three different REST services that will be used by the Alexa Skill: one service is used for getting the ID/s of messages that have not been played yet, the second one takes as input an ID and returns the associated audio file, and the last one takes an ID and deletes the associated message.

5.0.2 The Alexa Skill. Through the Alexa Skill Kit, developers can define new vocal functionalities by accessing external services. To demonstrate the effectiveness of the proposed solution, we developed a proof of concept Alexa Skill, *Petal notification*, which is able to connect to the Rule Manager, get the vocal message(s) and play it (them). The skill has also been certified. We defined the skill's logic as an AWS Lambda [11] Function in which, first, we identify the user who is interacting with the Echo device by allocating an ID to each user in the system. Next, a request will be sent to the Rule Manager to get the IDs of vocal messages related to the user. Later, a request for each retrieved ID will be sent to receive the corresponding audio file. As soon as an audio file is played, the vocal message will be deleted from the server. When the skill is activated, it queries the Rule Manager to discover whether there are vocal messages that are not yet played. If at least one message exists, it will be retrieved and played by the Echo device. This will be done by exploiting the audio tags specified in the Speech Synthesis Markup Language (SSML).

[9]https://developer.amazon.com/it-IT/docs/alexa/custom-skills/speech-synthesis-markup-language-ssml-reference.html#audio

[10]https://ffmpeg.org/

[11]https://aws.amazon.com/lambda/

5.1 Skill Activation

The Amazon Alexa Skills are in general passive: users should explicitly issue voice commands, by uttering a so-called wake word like "Alexa" followed by the invocation word associated to the skill (e.g., "Alexa, open Petal Notification"). This, for the users such as those who live with a disability, illness, or ageing could be difficult. In our project the target users are older adults, therefore we deemed useful to introduce additional ways to listen to the vocal reminders/alarms. In the integrated platform exploiting trigger-action rules, a vocal notification can be an action contained in a rule whose trigger(s) are activated when specific events occur in the user contest: in this case, the notification is played automatically, without any user explicitly asking for it.

In order to solve this problem, we exploited the OH Echo binding, which gives the possibility to start an *Alexa Routine* programmatically. An Alexa routine allows users to bundle together and automate several actions (e.g., activate a skill) using a single trigger or voice command (e.g., voice, time, location). Thus, through a routine, it is possible to launch a Skill whenever an event recognized by the Alexa ecosystem occurs. For instance, it is possible to set a routine so that when the user says, "I'm leaving", Alexa responds, "Have a good day" and then turns off all the lights. Thus, we defined a routine that launches the *Petal Notification Skill*, but, instead of expecting the user to say a specific voice utterance, it activates the routine in OH whenever the Rule Manager Add-On receives an action involving a reminder/alarm that is in a language not currently supported by Alexa. In this way, the skill is activated as soon as the rule is triggered by any relevant contextual event (for example, the user has not taken a pill in a certain period of the day), and the user can immediately listen the content of the reminder/alarm action defined in the triggered rule.

6 DISCUSSION AND CONCLUSIONS

In this paper, we have reported our initial experience in integrating a widely used vocal assistant (Alexa) in a personalization platform that allows end-users without programming experience to directly indicate how they want to exploit the variety of sensors and connected objects that are increasingly available to them. The goal is to better exploit the immediate and natural way of interaction supported through such a device by connecting it with events that occur in daily life and can be detected with the support of IoT technology. The initial target users of this integrated platform are older adults living autonomously in their home, and their formal or informal caregivers who know well their habits and abilities and can in this way create the rules defining the most suitable automation for their daily life. We have discussed an initial set of possibilities obtained through such integration, which extend the traditional way to use the vocal assistants. Besides, we have also indicated a technical solution to address one further limitation in Alexa concerning the languages currently supported. Such a solution allows the platform to generate vocal messages associated with some triggers in languages not supported by Alexa, thus lowering the associated language barrier generated by such limitation.

Future work will be dedicated to gathering empirical feedback on the integrated platform by the target users. For this purpose, we have planned several trials in various European countries in which the personalization platform (together with the associated sensors and the connected objects) will be deployed in their homes.

ACKNOWLEDGMENTS

This work has been partly supported by the AAL Project PETAL[12].

REFERENCES

[1] amazon. 2020. *Cleo*. Retrieved April 24, 2020 from https://www.amazon.in/Amazon-Cleo/dp/B01N5QDE0Y
[2] Tawfiq Ammari, Jofish Kaye, Janice Y. Tsai, and Frank Bentle. 2019. Music, Search, and IoT: How People (Really) Use Voice Assistant. *ACM Trans. Comput.-Hum. Interaction* 26, 3 (2019), 17:1–17:28.
[3] Giuseppe Desolda, Carmelo Ardito, and Maristella Matera. 2017. Empowering End Users to Customize Their Smart Environments: Model, Composition Paradigms, and Domain-Specific Tools. *ACM Trans. Comput.-Hum. Interact.* 24, 2, Article 12 (April 2017), 52 pages. https://doi.org/10.1145/3057859
[4] Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. 2017. Personalization of Context-Dependent Applications Through Trigger-Action Rules. *ACM Trans. Comput. Hum. Interact.* 24, 2 (2017), 14:1–14:33. https://doi.org/10.1145/3057861
[5] Manuel Jesús-Azabal, José Agustín Medina-Rodríguez, Javier Durán-García, and Daniel García-Pérez. 2019. Remembranza Pills: Using Alexa to Remind the Daily Medicine Doses to Elderly. In *International Workshop on Gerontechnology*. Springer, 151–159.
[6] Manuel Jesús-Azabal, Javier Rojo, Enrique Moguel, Daniel Flores-Martin, Javier Berrocal, José García-Alonso, and Juan M Murillo. 2019. Voice Assistant to Remind Pharmacologic Treatment in Elders. In *International Workshop on Gerontechnology*. Springer, 123–133.
[7] Henry Lieberman, Fabio Paternò, Markus Klann, and Volker Wulf. 2006. *End-User Development: An Emerging Paradigm*. Springer Netherlands, Dordrecht, 1–8. https://doi.org/10.1007/1-4020-5386-X_1
[8] Marco Manca, Fabio Paternò, Carmen Santoro, and Luca Corcella. 2019. Supporting end-user debugging of trigger-action rules for IoT applications. *International Journal of Human-Computer Studies* 123 (2019), 56–69.
[9] Kenneth Olmstead. 2017. *Nearly half of Americans use digital voice assistants, mostly on their smartphones*. Retrieved April 24, 2020 from https://www.finchannel.com/world/america/70314-nearly-half-of-americans-use-digital-voice-assistants-mostly-on-their-smartphones
[10] Stefania Pinnelli and Andrea Fiorucci. 2015. Giving voice to images: audio description and visual impairment: technological solutions and methodological choices. In *Ambient Assisted Living*. Springer, 347–355.
[11] Juniper Research. 2018. *Voice Assistants used in smart homes to grow 1000%, reaching 275 million by 2023, as Alexa leads the way*. Retrieved April 24, 2020 from https://www.juniperresearch.com/press/press-releases/voice-assistants-used-in-smart-homes
[12] Dan Robotham, Safarina Satkunanathan, John Reynolds, Daniel Stahl, and Til Wykes. 2016. Using digital notifications to improve attendance in clinic: systematic review and meta-analysis. *BMJ open* 6, 10 (2016), e012116.
[13] Yoshinori Shiga, Jinfu Ni, Kentaro Tachibana, and Takuma Okamoto. 2020. *Text-to-Speech Synthesis*. Springer Singapore, Singapore, 39–52. https://doi.org/10.1007/978-981-15-0595-9_3
[14] Graham Templeton. 2020. *Language Support in Voice Assistants Compared (2019 Update)*. Retrieved April 24, 2020 from https://www.globalme.net/blog/language-support-voice-assistants-compared#Alexas_Language_Support
[15] Milka Trajkova and Aqueasha Martin-Hammond. 2020. "Alexa is a Toy": Exploring Older Adults' Reasons for Using, Limiting, and Abandoning Echo. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376760
[16] Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. 2014. Practical Trigger-Action Programming in the Smart Home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 803–812. https://doi.org/10.1145/2556288.2557420

[12]http://www.aal-petal.eu/