A Flexible and Scalable Social Robot Architecture Employing Voice Assistant Technologies

Ruben Alonso
ruben.alonso@r2msolution.com
R2M Solution
Pavia, Italy

Emanuele Concas
conema@unica.it
University of Cagliari
Cagliari, Italy

Diego Reforgiato Recupero
diego.reforgiato@unica.it
University of Cagliari
Cagliari, Italy

ABSTRACT

In this paper a flexible and scalable architecture for Human-Robot Interaction is presented. The architecture allows embedding voice assistant technologies that can trigger different kinds of applications for the interaction between the robot and the user. The robotic platform can host applications not computationally expensive, otherwise they can be run on external computing devices or cloud. Zora, an existing robotic platform based on NAO, has been chosen as the platform where to develop applications on top of it. As voice assistant tool we have employed Google Assistant: its SDK and APIs have been leveraged for some of the applications we have deployed within our architecture. Therefore, the resulting Human-Computer Interaction has two main benefits: on the one hand the user may interact with any of the applications created on top of the robot; on the other hand, the user can also rely on voice assistant tools to receive answers in open-domain.

CCS CONCEPTS

• **Information systems** → **Computing platforms**; • **Human-centered computing** → *Human computer interaction (HCI).*

KEYWORDS

Humanoids Robot, Cloud Robotics, Language Understanding, Human-Robot Dialogue, Voice Assistant Technology

1 INTRODUCTION

The adoption of robots have gradually shifted in the last years from their employment within big manufacturing industries to more common places where the daily life is carried out, such as schools, people' homes, hotels, restaurants, etc. Furthermore, voice assistants technology has improved much giving a strong impact and leading to several changes in every day life. Examples of advanced and state-of-art digital voice assistants are the following: Google Assistant[1], Apple's Siri[2], Microsoft's Cortana[3], Amazon's Alexa[4], Wit.ai[5] and Snips.ai[6], an open source and privacy oriented solution. Generally, each of the tool above makes available its own SDK and APIs so that developers can build their own applications and distribute them (similarly as already seen in the last years with smartphones application on different devices, e.g. IOS, Android). In this way users have a wide set of applications they can download and use with their digital assistant. Clearly, all this brings new business opportunities in diverse areas.

The education is one of those. Besides the digital assistants, robots are being constantly and widely used in this domain [3] and STEM[7] in general [9]. For example, one robot used within the STEM domain is Elias[8], a social robot that helps students learning foreign languages. Another examples is represented by Makeblocks' mBot[9], which teaches kids to code. One more domain where the usage of robots is increasing is the health-care. One example in this domain is represented by robots used as socially assisting entities playing an important role for children with autism spectrum disorders [6] and for the active aging [5], current hot topic with an enormous economic and social impact.

Within such a topic, current research is carried out to program robots to help the assistance and rehabilitation of people. This alleviates care centers' workload and improves the efficiency of the treatments bringing to a win-win situation that results in a reduction of the costs for both the care centers and the patients[10].

Obviously, it is key that robots are well accepted by people. To comply with that statement, robots need to have characteristics and aspects that will permit humans to enjoy interacting with them keeping high the level of engagement. It is

Copyright ©2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1]https://assistant.google.com/
[2]https://www.apple.com/siri/
[3]https://www.microsoft.com/en-in/windows/cortana
[4]https://developer.amazon.com/alexa
[5]https://wit.ai/
[6]https://snips.ai/
[7]Science, Technology, Engineering and Mathematics
[8]http://www.eliasrobot.com/
[9]https://www.makeblock.com/steam-kits/mbot/
[10]https://bit.ly/2Xkro1C

CEUR-WS.org/Vol-2724/paper10.pdf

not an easy task as high levels of acceptance and engagement of the users are required in order to create a sense of affinity. One direction is to develop human-like appearance of the robot that should result in a more naturally interaction [11]. To complicate things is the following research outcome. One may think that a robot that perfectly looks like a human would highly engage users. In reality, once that he/she realizes that what looked real is indeed artificial, it is likely that the person loses its sense of affinity and the machine becomes uncanny [8]. To address this further challenge, a lot of robot manufacturers aim at designing humanoid robots with improved fluid movements and equipping them with several sensors to let robot's appearance and interaction become less distinguishable from a human being.

Pushing toward this direction, in this paper we introduce a flexible and scalable architecture for Human-Robot Interaction (HRI) that employs voice assistant technologies. Such an architecture allows the robots to unlimited (software and hardware) extensions of their interaction with the users. In fact, one of its main advantages is that it is possible to embed and integrate any external software framework needing high computational power and that can reside on cloud systems or external servers leaving the robot's resources free and focused to the interaction with the users. The resulting interaction is therefore more fluid and efficient.

The reader notices that voice assistant technology can also understand people with severe speech impediments[11] giving therefore to the HRI a further boost. To close the circle, as already mentioned above, this kind of technologies provides SDK and APIs, useful to developers to create new skills or actions that can augment their effectiveness for specialized interactions. Examples might be buying groceries or booking a ride[12]. This makes digital assistant technologies even more effective and their widespread is therefore growing. It is enough to reflect to the fact that the number of voice assistant skills and actions is more than duplicated in 2019: only in the USA the Google Actions grew up of a factor of 2.5, for a total of 4253 actions whereas Amazon counts for more than 80000 Alexa Skills worldwide [12].

The robotic platform where the proposed architecture has been developed on top of it is shown in Section 2. Section 3 describes the proposed architecture and what type of applications can be included and integrated. Finally, Section 5 ends the paper with conclusions, ideas and directions where we are headed.

[11]https://www.voicesummit.ai/blog/how-voice-tech-is-slowly-including-people-with-speech-impediments

[12]for a list of current Google Actions check https://assistant.google.com/explore/

2 THE ROBOTIC PLATFORM

We have employed Zora robotic platform for our architecture[13]. Zora is basically a NAO robot of Softbank Robotics[14] extended by Zora Robotics[15] with a middleware software layer that allows everyone without programming skills to program different behaviours of the robot. Moreover it is even possible to completely control the robot in all of its aspects, to move the robot and use all its sensors, to change the network settings, set the language (there are eight different options) and several other configurations. The Zora Composer is one important element of the middleware layer. It allows creating applications and behaviours for Zora by performing drag and drop actions of boxes from a menu into a given timeline. The composer can be used by non technical people and it is also possible to use its web version. It follows the same rationale of Scratch[16].

3 ARCHITECTURE

The architecture for social robotics we propose in this paper consists of four main components, detailed in the following sections. Figure 1 depicts them and the high-level design of the architecture.

HRI Component (HRIC)

On top of the scheme there is the HRI Component (HRIC). It handles the interaction of the robot with the users. One can load several HRICs into the robot, for different tasks (e.g. control its behaviours, have access to its sensors and parts). HRICs consist usually of lightweight programs. Reason is because the hardware resources of the robotic platform should be left free for the robot itself and further computations should be as light as possible. Clearly, a certain robotic application may be entirely developed through an HRIC as long as it does not require heavy computational power or demanding storage capabilities. Each application in the proposed architecture should always have a *control HRIC*. This acts as a master program and decides the other HRICs to run, depending on the interaction with the user. E.g. if the user says *play Bingo*, the robot, after the execution of the Voice Assistant Component, calls the HRIC responsible for the Bingo game through the *control HRIC*.

[13]https://www.youtube.com/watch?v=lO52sLF-u_4&t=1s

[14]https://www.softbankrobotics.com/

[15]http://www.zorarobotics.be/index.php/en/

[16]https://scratch.mit.edu/

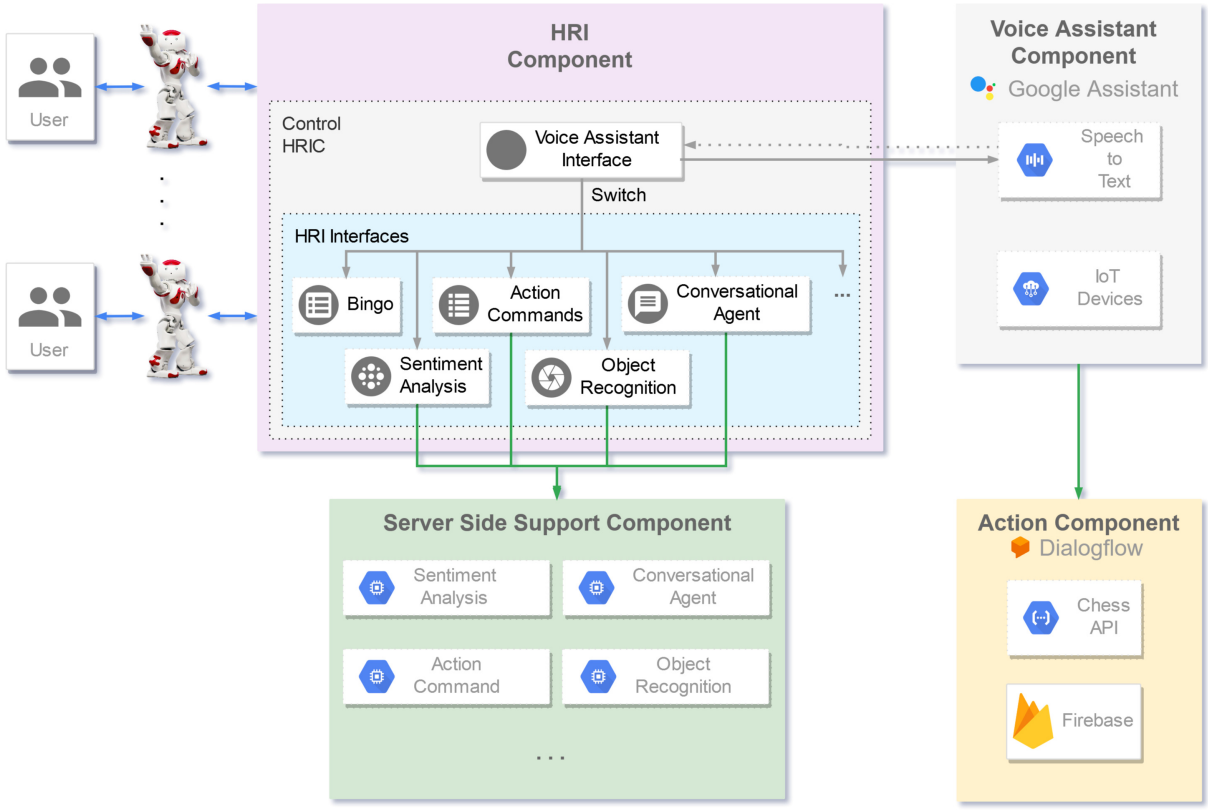

Figure 1: Proposed high-level architecture. The HRIC interacts with the user and sends the user's audio to the VAC. The VAC decides which modules to trigger. It sends to the HRIC either a string indicating which HRIC needs to be called or the result of the voice assistant cloud (if the user asked something in open domain). Every application includes an HRIC, uploaded into the robot. An associated SSSC can be present, uploaded into the cloud and that can be shared among several robots.

There are tree items the control HRIC must contain:

(1) a voice assistant interface: this is an abstraction layer of the employed voice assistant tool. It contains the tools to elaborate the audio and send it to the cloud used by the assistant tool to later retrieve the output. If the user invokes an application, then the calling string (returned by the speech-to-text tool of the voice assistant tool) is sent out of the box to the *Switch application*.
(2) Switch application: when the user invokes an existing application, the returned string from the speech-to-text tool of the voice assistant component is forwarded to a switch element that routes the request to the HRIC of the called application.
(3) HRIC interfaces: these are directly connected to the switch element and to the HRIC of the different developed applications. When the called HRIC ends its tasks, the control is returned back to the voice assistant interface.

An example of a control HRIC is displayed in Figure 2.

Voice Assistant Component (VAC)

The key component of the proposed architecture is represented by the VAC. Its main function is to receive the audio stream of the user from the control HRIC and provides output according to certain actions. At technical level, it is responsible of detecting the end of the user speech. The VAC speech-to-text engine processes the received audio. If the user input text triggers one of the applications previously loaded into the robotic platform then the underlying application is started by the control HRI. If the user text does not trigger any applications, then the processed user text is sent to the voice assistant cloud which replies depending on the adopted voice assistant tool. Thus the user can either start applications loaded into the robot or interact with the voice assistant tool through the robot in open domain. The VAC is general and can therefore employ any of the existing voice assistants tools such as Google Assistant, Amazon Alexa, Apple Siri, Microsoft Cortana, Wit.Ai, and so on. For a perfect integration of the voice assistant tool with our architecture it should have the following capabilities:

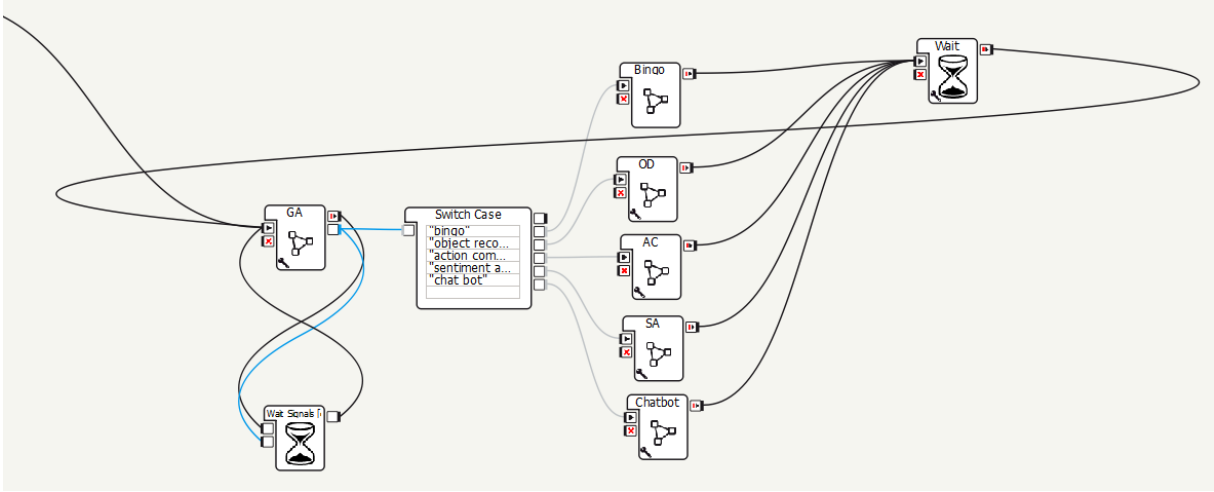

Figure 2: An implementation of the control HRIC using the Choregraphe suite. The first hourglass (⧖) on the left is needed because the voice assistant interface should not be triggered if it has started an HRIC. The second hourglass on the right is needed only to make more effective the interaction between the robot and the human.

- it is possible to send user's audio by the given robotic platform to the voice assistant using its SDK/APIs;
- the output to the user's statement or questions are returned in a audio or text format. In the first case the robot will simply play it; in the second case the robot will text-to-speech the returned output;
- the voice assistant tool includes its APIs and can thus be extended with custom script commands. These will not forward the user text to the voice assistant but will trigger developed applications within the voice assistant tool that can be further configured and changed.

Server Side Support Component (SSSC)

As already seen for smart phones applications, developers should create very light applications to be uploaded directly into the robotic platforms and keep the heavy calculus on servers or cloud. Therefore, the chosen robotic platform should not be loaded with deep learning algorithms to be trained nor loaded with gigabytes of data. This observation brings to an architecture pattern to follow known as cloud robotics. Therefore in our proposed architecture we have included the Server Side Support Component (SSSC). It consists of all the high-demanding resources (CPU, GPUs, memory, storage) software needed for a certain application. The SSSC runs in external servers or within the cloud. A given HRIC is able to communicate with its corresponding SSSC through REST APIs exposes by the latter. They can be called via HTTP requests directly from the HRIC loaded and running into the robot. A given SSSC must have an associated HRIC for communication of data and results for a particular task. Each SSSC can run independently from each other and from the other components present in the architecture. Different SSSC can run in parallel in servers or cloud. Although many HRICs can be loaded into the robotic platform only one can be currently active, triggered by the VAC through specific voice commands. With such a flexibility, all the cutting-edge technologies in fields such as Computer Vision, Artificial Intelligence, Big Data, Semantic Web, Natural Language Processing, together with new hardware including GPUs to speed up computation of several machine learning tasks, can be efficiently employed for an effective HRI.

Action Component (AC)

Voice assistant tools functionalities can be extended by actions. Through actions it is possible to make more effective the resulting HRI by creating personalized interactions for users. Examples might include turning lights on/off using plugged in sensors, playing games, booking a restaurant or flight and so on.

Differently from SSSCs, ACs do not need to have associated HRICs unless they need access to the robot sensors or parts other than robot's speakers and microphones. The reason is because actions reside on the voice assistant cloud and can be triggered directly from the VAC by giving explicit commands to the voice assistant. To show one example, we have developed one AC on the use case which instantiates our proposed architecture. It is called when the user says to the robot *I want to play with Mr. Chess*. The VAC, therefore, triggers the Chess game we have developed on the voice assistant cloud. The AC provides one more layer for scalability and flexibility. In fact, all the actions created for the employed voice assistant tool can be used out-of-the-box

with our architecture. The result is that our architecture can exploit the unlimited number of actions that are being developed for voice assistant tools.

4 THE PROPOSED USE CASE

For the HRIC we have adopted the Choregraphe suite[17]. The Voice Assistant Interface (VAI), a peculiar interface used to communicate with the voice assistant platform, includes a thread which records the audio coming from the robot's microphone using the command-line sound recorder for ALSA[18] drivers with the *arecord* command and the output is continuously sent to a middle-ware layer that takes care of buffering and that sends the audio to the voice assistant platform. Through the middle-ware, the VAI receives and parses the response sent by the voice assistant platform. The switch is a simple module which routes the VAI output and starts the desired application of the user.

The VAC module employs Google Assistant as voice assistant tool. The Assistant Embedded APIs[19] are used for sending the audio to the VAC. Google Assistant can be extended with Custom Device Actions[20] which allow the robot to have special abilities not covered by the default GA's traits[21]. There should be a custom action for every callable and executable HRI interface.

ACs are built with Actions on Google[22]. Actions On Google works in collaboration with Dialogflow, a user-friendly tool provided by Google, that allows using machine learning to understand the natural language of what users are saying. [1, 2, 4, 7, 10] represent successful applications developed on top of our use cases.

5 CONCLUSIONS

In this paper we have presented a general, flexible and scalable social robot architecture which uses voice assistant technologies. We also showed the use case we have developed using Zora as robotic platform and Google Assistant as voice assistant technology. The limitations and issues of our architecture are strictly connected to the chosen robotic platform. For example, for the proposed use case, a limitation is represented by the low-quality microphones and the native speech-to-text tool that not always correctly identifies the spoken text. The use case architecture and the applications we have developed can be seen at https://bit.ly/2tkRhkc. The source code is hosted in repositories accessible from http://hri.unica.it.

[17] http://doc.aldebaran.com/2-4/software/choregraphe/index.html
[18] http://alsa-project.org/
[19] https://developers.google.com/assistant/sdk/reference/rpc/google.assistant.embedded.v1alpha2
[20] https://developers.google.com/assistant/sdk/guides/library/python/extend/custom-actions
[21] https://developers.google.com/assistant/sdk/reference/traits/
[22] https://developers.google.com/actions/

REFERENCES

[1] Mattia Atzeni and Diego Reforgiato Recupero. 2018. Deep Learning and Sentiment Analysis for Human-Robot Interaction. In *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers*. 14–18. https://doi.org/10.1007/978-3-319-98192-5_3

[2] Gianluca Bardaro, Danilo Dessì, Enrico Motta, Francesco Osborne, and Diego Reforgiato Recupero. 2019. Parsing Natural Language Sentences into Robot Actions. In *Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26-30, 2019*. 93–96. http://ceur-ws.org/Vol-2456/paper24.pdf

[3] Tony Belpaeme, James Kennedy, Aditi Ramachandran, Brian Scassellati, and Fumihide Tanaka. 2018. Social robots for education: A review. *Science Robotics* 3, 21 (2018). https://doi.org/10.1126/scirobotics.aat5954 arXiv:https://robotics.sciencemag.org/content/3/21/eaat5954.full.pdf

[4] Amna Dridi and Diego Reforgiato Recupero. 2017. Leveraging semantics for sentiment polarity detection in social media. *International Journal of Machine Learning and Cybernetics* (19 Sep 2017). https://doi.org/10.1007/s13042-017-0727-z

[5] Grazia D'Onofrio et al. 2017. MARIO Project: Experimentation in the Hospital Setting. In *Ambient Assisted Living - Italian Forum 2017, eighth Italian on Ambient Assisted Living Forum, ForItAAL 2017, 14-15 June, 2017, Genoa, Italy*. 289–303. https://doi.org/10.1007/978-3-030-04672-9_20

[6] David Feil-Seifer and Maja Mataric. 2008. Robot-assisted therapy for children with autism spectrum disorders. In *Robot-assisted therapy for children with autism spectrum disorders*. 49–52. https://doi.org/10.1145/1463689.1463716

[7] Federica Gerina, Barbara Pes, Diego Reforgiato Recupero, and Daniele Riboni. 2019. Toward Supporting Food Journaling Using Air Quality Data Mining and a Social Robot. In *Ambient Intelligence - 15th European Conference, AmI 2019, Rome, Italy, November 13-15, 2019, Proceedings*. 318–323. https://doi.org/10.1007/978-3-030-34255-5_22

[8] Masahiro Mori, Karl MacDorman, and Norri Kageki. 2012. The Uncanny Valley [From the Field]. *IEEE Robotics & Automation Magazine* 19 (06 2012), 98–100. https://doi.org/10.1109/MRA.2012.2192811

[9] Marievie Panayiotou and Nikleia Eteokleous. 2017. *ROBOTICS AS MEANS TO INCREASE STUDENTS' STEM ATTITUDES*. InScience Press, Chapter Projects and Trends, 216–220.

[10] Diego Reforgiato Recupero, Danilo Dessì, and Emanuele Concas. 2019. A Flexible and Scalable Architecture for Human-Robot Interaction. In *Ambient Intelligence - 15th European Conference, AmI 2019, Rome, Italy, November 13-15, 2019, Proceedings*. 311–317. https://doi.org/10.1007/978-3-030-34255-5_21

[11] Diego Reforgiato Recupero and Federico Spiga. 2019. Knowledge acquisition from parsing natural language expressions for humanoid robot action commands. *Information Processing & Management* (2019), 102094. https://doi.org/10.1016/j.ipm.2019.102094

[12] Voicebot. 2019. Google Assistant Actions Total 4,253 in January 2019, Up 2.5x in Past Year but 7.5% the Total Number Alexa Skills in U.S. https://voicebot.ai/2019/02/15/google-assistant-actions-total-4253-in-january-2019-up-2-5x-in-past-year-but-7-5-the-total-number-alexa-skills-in-u-s/