# SYSTEM SOFTWARE ARCHITECTURE FOR ADVANCING HUMAN-ROBOT INTERACTION BY CLOUD SERVICES AND MULTI-ROBOT COOPERATION

*Anna Lekova*[1], *Paulina Tsvetkova*[1,3], *Anna Andreeva*[1,2], *Miglena Simonska*[2] *and Adelina Kremenska*[1]

(1) Institute of Robotics, Bulgarian Academy of Sciences; (2) South-West University, Blagoevgrad; (3) University of Library Studies and Information Technologies Bulgaria

* Corresponding Author, e-mail: a.lekova@ir.bas.bg

**Abstract:** Human-like interactions with robots based on Conversational AI facilitate assistance and teamwork in various contexts. Those interactions are further enhanced by utilizing physical presence and context from the robot's hardware. Robot cooperation is also especially useful, when software or hardware resources have to be shared in a multi-robot system. Therefore, we propose a modular software architecture for multi-robot cooperation that extends the integration of Conversational AI into Socially Assistive Robots, previously suggested by authors. It utilizes a flow-based approach that involves shared repositories and direct or message-driven communication to convey natural language transcriptions among robots in order to support their cooperation. By experiments we evaluated the cooperation between NAOqi based robots and Furhat robot. Our experimental results demonstrate architecture's modularity and adaptability to different cloud services, along with its effectiveness for interactions involving multiple robots.

**Key words:** conversational artificial intelligence; socially assistive robots; NLP cloud services, multi-robot cooperation, furhat robot.

## 1. INTRODUCTION

In today's technologically advanced society the smooth incorporation of Conversational Artificial Intelligence (ConvAI) into our daily lives has opened avenues for more intuitive and personalized human-robot collaborations. Human-like interactions with robots, based on ConvAI, facilitate assistance and teamwork in various contexts. Those interactions are further enhanced by utilizing physical presence and context from the robot's hardware. Robot cooperation is also especially useful, when software or hardware resources have to be shared in a multi-robot system. Artificial Intelligence (AI) significantly contributes to this context, as the integration of Natural Language Processing (NLP) with machine and deep learning in ConvAI facilitates natural conversations. AI serves as a tool on various platforms, such as the one outlined in [1], to address both educational and professional needs. ConvAI can be used in virtual settings, however it becomes more beneficial when it is integrated in SARs that can interact with humans physically. The robots use diverse sensors and movement controllers to understand the environment around them, which provides customized help to users in different actions and tasks, increased motivation and engagement, personalized vocabulary for the particular use case or specific management of user presence and attention. The ConvAI, integrated in robots, has to generate adequate responses via analysing verbal expressions and taking into account the context of the conversation. An example of this is when the Furhat robot speaks, its face is animated and shows feelings and expressions that fit the meaning and tone of the words. (https://furhatrobotics.com). Thanks to its own animated and contextually relevant text-to-speech, Pepper robot (https://www.aldebaran.com/en/pepper) interacts with people naturally and engagingly. Nao robot (https://www.aldebaran.com/en/nao) offers a buddy-like interaction with children, featuring cute animations.

A modular system software architecture should be designed, implemented and evaluated in such a way to facilitate the utilization of various sensory and actuator systems in robots. This will ensure adaptability, reliability and testability for different operational systems of SARs. We extended additionally the proposed by us in [2] modular software architecture that enhances human-robot interaction by ConvAI, for multi-robot cooperation. The architecture splits the programming software into two layers - high and low. The high-level layer connects the cloud NLP services with the low-level layer and the low-level layer connects the hardware (sensors and actuators) with the high-level layer. Robots can subscribe to services that can provide and request information since the architecture maintains both request-response and publish-subscribe mechanisms. By experiments we evaluated the cooperation between the robots NAO, Pepper and Furhat. Our experimental results demonstrate the modularity and adaptability of the software design to different NLP cloud services and various multi-robot cooperation scenarios. We provide evidence through a successful proof case for involving ConvAI integration in Furhat robot. A software architecture was implemented to integrate NLPcloud, IBM Watson and OpenAI NLP services with Furhat native software using NodeRED as a platform. The flow-based approach of NodeRED was tested to validate the cooperation among robots. This included the exchange of natural language transcriptions between Furhat and the NAOqi-based robots, enabling teamwork during speech and language rehabilitation.

To the best of our knowledge, we first propose a generic software architecture designed to facilitate active and real-time cooperation among socially-assistive robots, aiming to enhance human-robot interaction. This allows low- resourced or low-intelligent SARs to benefit from cooperation with more advanced robots. The advantage of NAO and Pepper is their humanoid form, whereas Furhat, being a conversational robot, has an embedded framework for structured dialogue. Additionally, the proposed architecture also enables human-like interactions through transcriptions among the robots themselves.

The rest of the paper is organized as follows: Section 2 presents related works, Section 3 – the design and implementation of the proposed software architecture for multi-robot cooperation, while in Section 4 are discussed several cooperation scenarios among three SARs. Then a conclusion follows.

## 2. RELATED WORKS

Social robotics is growing rapidly and is emerging as a key area of study in the era of Information Technologies [3]. NAO has gained popularity in the area of child-robot interaction research because of its low cost and extensive range of features, making it one of the most commonly employed social robots. Pepper is another humanoid robot, which has body language, move around and analyse human's expressions and voice. The robot possesses high-level interfaces for multimodal communication with the people around it. Pepper has been used in research and development applications involving individuals of various age groups, ranging from children to elderly people. Furhat is a social robot with human-like expressions and advanced conversational AI capabilities, such as listening, speech, gestures and attention to users' attendance. Currently, it is perceived as the world's most advanced social robot. The use of conversational AI-integrated agents is a new area, with only a couple of studies published so far [2, 4÷7]. The authors in [2] proposed a generic software design for integrating ConvAI into SARs, employing a flow-based method of NodeRED. Two experiments were conducted to assess the architecture's applicability. The first experiment evaluated diverse NLP services of different providers (NLPcloud, IBM Watson, MS Azure, Google and OpenAI), whereas the second experiment examined the incorporation of ConvAI for specific tasks (speech recognition and text classification) with NAO's and Pepper's native software. Different works have studied the application of large language models in social robots in Human-Robot Interaction (HRI). An example for such collaboration is a story-telling game CreativeBot [8] where a child and a robot Furhat create a story via turn-taking. On the other hand, the authors in [9] examine the difficulties associated with utilizing large language models for multi-modal open-domain dialogues, which derive from the interactions between older adults and a personalized companion robot - the Furhat robot, powered by GPT-3.5. Authors in [7] use the Pepper robot for chit-chatting and enhance embodied dialogs by providing additional 'cloud-background' dialogue capabilities to complement the preexisting natural language understanding ability.

Researchers have investigated how ConvAI in robots can help students with learning problems in education [2, 3]. Humanoid robots with the capability of speech interaction for children with language and speech difficulties or Autism Spectrum Disorder (ASD) have achieved outcomes similar to those of human tutoring on restricted tasks. In [3] the authors talk about the potential of social robots to serve roles in education, acting as tutors or peer learners. Papers [4÷8] present different approaches to improve the capacity of social robots and provide explanations of the proposed software system. They talk about the pros and cons of using cloud services for social robots, such as issues related to safety and confidentiality and network speed. However, the framework needs more research and testing in actual settings. The authors in [5] debate over standards and utilization of a cloud-based robot system for long-term interaction. The authors developed a cloud-based and modular system called "Personal Assistant for a Healthy Lifestyle". It has a long-lasting human-robot communication that relies on mixed AI and shared knowledge to guide the interaction. It also proposes some potential improvements that can be added to the system, such as GPS tracking, eye-gaze tracking and advanced speech recognition. Deuerlein et al. [6] examined how precise and fast cloud-based speech-recognition systems (SRS) are in human-robot communication. Usually, the time latency for cloud-based SRS can vary from milliseconds to several seconds. They found that the quality of the network connection and the cloud server's computing power greatly affect the precision and delay of cloud-based SRS.

While multi-robot cooperation is common in industrial applications, this is not the case in Social Robotics. After conducting a literature review for cooperation among SARs, we found only one article discussing a socially-assistive robotic system called SAR-Connect [10], which combines the benefits of a physically present social robot and virtual reality (VR). The robot plays a complex role in monitoring, providing feedback and guiding users in meeting physical, cognitive and social requirements during VR-based tasks with older adults. While the VR task provides diverse stimuli, the platform does not currently incorporate cooperation between robots. Therefore, we propose a software architecture for active and real-time cooperation between socially-assistive robots for advancing human-robot interaction. This allows low- resourced or low-intelligent SARs to benefit from cooperation with more advanced robots, such as Furhat.

After summarizing the challenges addressed in the reviewed studies, we concluded that the most important are the technical issues in creating AI Conversational interfaces, encompassing natural language processing, speech recognition, text generation and timely text-to-speech synthesis. Additional challenges included multi-robot cooperation, user acceptance, the absence of evidence-based practices and the need for further research on effectiveness and personalization of AI conversational interfaces. Furthermore, there aren't any methodologies for integrating ConvAI in robotic systems, along with ethical and legal concerns. Our goal is to enhance the conversation with SARs using NLP and to overcome the technical issues related to smooth software integration for ConvAI and multi-robot cooperation. To achieve this, we suggest a software framework that is based on data flows. We employ Node-RED (https://nodered.org/), a platform for programming with data flows, that allows publish-subscribe communication, which lets threads exchange messages without waiting for each other. Flow-based programming is preferred because it is a modular method that uses separate components, such as nodes and flows, which autonomously handling specific tasks. This makes the architecture software-layer independent and flexible for different SARs and NLP scenarios, enabling SARs to talk through natural language texts.

## 3. SOFTWARE ARCHITECTURE FOR ROBOT COOPERATION

This section begins by presenting successful proof case for the integration of ConvAI in the Furhat robot, highlighting the effectiveness of the generic software architecture proposed in [2]. Then, follows an explanation how Node-RED (the central component in the software architecture) facilitates robot cooperation.

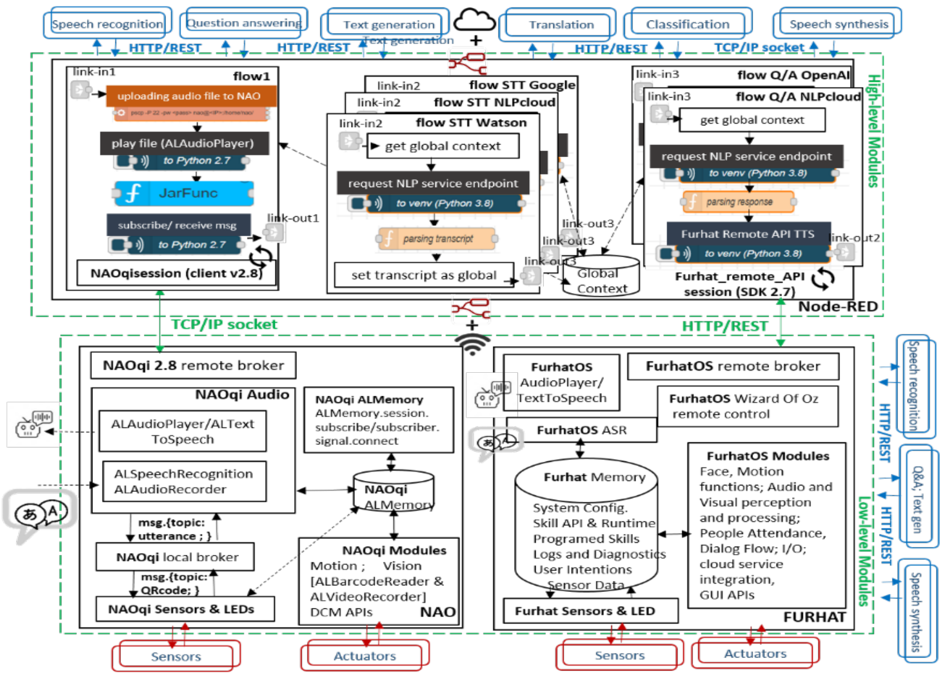


*Fig. 1. A system software architecture for cooperation among robots*

Figure 1 illustrates the integration of ConvAI in the Nao and Furhat robots through the implementation of the generic software framework suggested in [1]. This software architecture is based on two layers of data flow and uses common storage, direct or message-oriented communication channels, and request-response or publish-subscribe patterns. The first layer is implemented with Node-RED. It is closer to interactions between users and robots and manages high-level flows for text or speech processing for Conversational AI via RESTful interfaces to NLP cloud services. The communication between the flows and APIs for cloud services is split into different nodes that manage the interactions, exchange messages and have their own input and output settings. Flows can work with both “stateless” and “stateful” architectures. Designated nodes in the flows interact with the shared repository in order to set or retrieve global context. Robots post or get natural language transcriptions in the shared repository in order to assist their teamwork. The second layer, closer to the robot OS and hardware, consists of low-level modules. Each layer has unique publish-subscribe mechanisms and shared repositories. Data and message exchange between layers is enabled through remote protocols. This lets external clients, such as Furhat, read and record data to the NAOqi-based robots from the high-level modules. Inter-layer communication involves nodes creating child processes to control modules through robot APIs, using TCP/IP sockets for Nao/Pepper or HTTP/REST protocols for Furhat.

We evaluated the modularity of the software architecture for the Furhat robot, examining its ability to access different NLP cloud services for text understanding and generation in the higher layer. This evaluation also considered the adaptability to various robot operating systems and hardware in the lower layer. Despite being a conversational robot, utilizes the upper layer for services that it cannot independently provide. Another advantage is for the users lacking programming in Kotlin. Developing Furhat skills using the Kotlin SDK can be bypassed by applying integration with NLP cloud services into Furhat native software using Node-RED and Furhat remote APIs. We performed tests with the Furhat robot, using the same high-level component and utilizing NLPcloud (GPT-J, GPT-NeoX), IBM Watson and OpenAI (GPT3.5-turbo) for various NLP tasks, including STT, TTS, Question Answering, Text Classification, Automatic Speech Recognition and Sentiment Analysis. The same nodes in Node-RED flows were used to parse transcripts and responses. A modification was implemented on the remote client in Node-RED for the Furhat Remote API session (Figure 1, upper right part).

Furhat Remote API is one of the programming methods for developers to create applications (or "skills") for the robot. The Furhat Remote APIs provide a way to connect and give commands to the robot from an external computer on the same network. The core robot I/O functionality, including listening, speech, gestures, attention, user attendance and LED control, can be accessed via an HTTP-based REST API using any programming language (with support for 50+ languages). There is a wrapper specifically for Python Remote API that simplifies our implementation for the Furhat robot. In Node-RED, custom flows communicate with NLP services in the cloud via RESTful APIs and different programming language clients. Python has the benefit of having many libraries. Using a python client is beneficial because it lowers the delay when connecting to cloud-based services that need data transfer, such as sending serialized audio file for Automatic Speech Recognition (ASR). There is no issue with robots using different Python versions. The NAOqi framework, used for coding NAO and Pepper robots, is accessed through Python 2.7 scripts using 'exec' or 'python_shell' nodes in Node-RED. Custom nodes in different flows can create a TCP/IP connection with the NAOqi broker to run 'qi' services or 'qi' applications. The result from the Python script is sent back as the payload message in Node-RED. We distinguished the various Python scripts by the label on their icons. In Figure 1, the node called 'python_shell' with the label 'to Python 2.7' connects to NAOqi OS that runs on Python 2.7. Conversely, the 'python_shell' node labeled 'to venv (python3.8)' was employed to access Furhat Remote API for TTS in virtual Python 3.8 environment.

The other programming methods for Furhat developers is by Kotlin. Kotlin Skill APIs provide full access to the Furhat Skill Framework and the platform's capabilities for building more complex interactions. A Kotlin skill is an application that controls the robot's interactions and behaviour, including elements such as dialogue structure with states, triggers and natural language understanding through intents and entities. Kotlin skills are uploaded in the Furhat memory. The deployed skill can be run from a terminal/console by using java -jar <name_of_skill_file>.skill or from Furhat web interface. By the implementation proposed in this study, Kotlin skills are started from the higher-level modules via a Node-RED node (node-red-contrib-jar-func). It is a function node that calls an executable .jar file (see Fig.1 - node JARFunc in flow1).

Node-RED enables cooperation among robots by acting as the main element for communication between them. The virtual 'link out' and 'link in' nodes on the high level facilitate subscription and publishing messages between robots in Node-RED. The 'link out' node can be set up to transmit messages to all linked 'link in' nodes and notify them of certain events. The 'link in' node gets messages from other flows and inserts them into the flow context. Through the flows in Node-RED, requests are made to NAOqi modules, allowing communication with the internal memory and hardware (DCM APIs) of NAOqi-based robots. Other flows make requests to Furhat remote APIs and hardware. To support robot cooperation we utilize publish-subscribe paradigm in both NAOqi broker and Node-RED. Furhat current SDK only employs publish-subscribe paradigm for graphical interfaces with Furhat skills and in order to ask for cooperation with the Nao/Pepper robots, Furhat has to subscribe remotely to their NAOqi Message Brokers (see flow1 in Fig.1). Furhat can subscribe to events, perform memory operations or initiate audio playback through remote calls to NAOqi APIs. Depending on the topic, various remote API clients are activated. They can perform read/write operations on specific data pairs in the robot's shared storage. On the other hand, in order Furhat to enrich Nao/Pepper robots, Furhat uses the 'link in' and 'link out' nodes in Node-RED. This allows Nao/Pepper to subscribe to Furhat services and to be notified, while Furhat publishes new information. Through remote calls to Furhat APIs, functionalities such as listening, speech, gestures, and attention to users' attendance can be accessed - features that are either absent or not as advanced in NAOqi robots.

Figure 1, and more specifically flow 1 in the higher layer, illustrates how Furhat is subscribed to Nao for audio playback cooperation. This is a practice when the robot TTS is not native for a specific language and an alternative is to use a cloud service to obtain an audio file from TTS and play it on the robot. Furhat comes with credentials for speaking voices like PollyVoice and Azure voices, however Bulgarian language is not supported with such voices. If personal Azure credentials are not available, NAOqi or Furhat Python remote APIs can be utilized to access gTTS (Google Text-to-Speech), a Python library and CLI tool that interfaces with Google Translate's text-to-speech API. This process involves writing spoken MP3 data to a file. Furhat can play an audio file from an URL, however audio files are cached on the robot in order to improve the execution of subsequent requests for the same dialog data. Until now, the clear cache feature is accessed through the Furhat robot's web console and not programmatically. Alternatively, the file can be played on the NAO side. Furhat needs to subscribe to event via NAOqisession in the high-level (see link-out1 in flow1). Via remote calls to NAOqi APIs for NAOqi Audio, the output message for topic1 in flow1 involves uploading a synthesized audio file to the robot using the PuTTY pscp network protocol.

An additional advantage of implementing the system software architecture for Furhat cooperation is Nao's opportunity to utilize cloud AI services deployed in the Furhat Skill Framework. This is evident in Fig.1 - the supplementary cloud NLP services to the right of the Furhat low-level layer. Thus, Furhat can cooperate to any other robot that has not embedded Conversational AI services. For instance, NAOqi-based robots can use these skills from a Node-RED flow by executing either curl node or jar-func node by calling an executable .jar file. Thus, Nao, via the Furhat Skill Framework and Node-RED, can access Wolfram|Alpha Computational Intelligence (https://www.wolframalpha.com/). The skill uploaded on the robot integrates the Furhat I/O with OpenAI's text completion service, including GPT-3 and ChatGPT. To connect with the OpenAI service, FurhatOS relies on skill dependencies - TheoKanning's Java library for utilizing OpenAI's GPT APIs.

## 4. IMPLEMENTATION AND ASSESSMENT OF THE SOFTWARE ARCHITECTURE FOR MULTI-ROBOT COOPERATION

In this section we describe and discuss several cooperation scenarios involving three SARs within diverse multi-robot cooperation frameworks. We also explore the design of various use cases in Speech and Language Therapy (SLT) that require cooperation with Furhat robot.

The human-like communication with children who have speech and language impairments starts either by utterance or QR code, shown to the upper camera of the NAO robot. Furhat robot need NAO cooperation in QR code reading, because presenting images to a humanoid robot is more intuitive in the SLT context. The speech recording is initiated by the child by touching a tactile sensor on the robot during the verbal interaction. This activates the robot's speech recognizer, which then starts the robot microphone and audio module to record the speech. The current state of the robot is reflected by the LED colours in its eyes: white means ready to listen, blue means listening and recording and green means processing. The following logic is applied to stop the recording: if there is silence for 3 seconds, i.e. the 'Speech Reco' box outputs are inactive, the recording is stopped. The audio file is saved in 'wav' format in the robot's internal memory. An event is activated by the PUB/SUB on the robot side and 'event signaling' is received at the higher level from the remote NAOqi 2.8 client. A video showing a possible interaction with the NAO robot can be seen on YouTube (https://youtu.be/uUagm9YHU8k). We experimented with the Pepper robot using the same high-level module and NLP cloud APIs by altering the remote client in Node-RED for NAOqi 2.5 and adapting the modules at a lower level for Pepper. By reading touch data from its head sensors, Pepper starts the recording of a question, unlike NAO, which relies on hand touch sensors. The robot uses the ALAnimatedSpeech module to speak with postures, gestures and custom animations embedded on the robot.

We compared the performance of both NAOqi and Furhat Python remote APIs for TTS. Both NAOqi and Furhat Python remote APIs can be used to access gTTS (Google Text-to-Speech). Nao faces latency issues while using GoogleTTS from the high-level modules. Although, Furhat uses the same Python libraries for Google ASR, file download, conversion and upload, Furhat exhibits no latency during TTS. This is a result of the limited resources of Pepper and Nao. Therefore Furhat robot can cooperate for voices that are native to Furhat and not to them. NAOqi robots can request Furhat to handle this from Node-RED flows and Python child process. Furhat will play an audio file uploaded on the web using its Python remote API. In general, it is important to prioritize the use of the local TTS module instead of cloud NLP services to enhance latency when the robot has an internal TTS module and the language is native,

Nao may request cooperation from Pepper or Furhat to utilize their tablets for visualizing pictures or words during blending and segmenting activities. Blending involves combining individual phonemes (speech sounds) to form a word. This skill is essential for reading. For example, blending the sounds /c/ /a/ /t/ results in saying the word "cat". Segmenting is the opposite of blending. It involves breaking down a word into its individual phonemes. This skill is crucial for spelling. For example, segmenting the word "dog" results in identifying individual sounds /d/ /o/ /g/. During the "Rhyming

Game”, Pepper cooperate with Nao or Furhat to display a word and alternatives for rhyming on Pepper's tablet (https://youtu.be/Wasb3sAL76Y). In interactive scenarios, Pepper supports speech therapists via fun and interactive game in which a child is given a word accompanied by various pictures. The goal of the game is the child to guess which pictures rhyme with the given word and to receive a corresponding feedback. The game is developed using HTML, CSS, JavaScript and the QiMessaging library, which is used to define functions for the robot. More information and the open-source code can be found at https://github.com/MaximHr/PepperRobot.

In interactive scenarios, Furhat not only assists robots but also supports speech therapists and families in intensive speech and listening exercises. A video demonstrates Furhat's teamwork in teaching children short poems through repeated strophes (https://youtu.be/MLkateQay-M). This Furhat skill, deployed either on a virtual or physical robot, can be accessed by all robots connected via the proposed software architecture. Furhat skills designed for intensive speech and listening therapy in children with stuttering can aid parents of stuttering children who own a Nao robot. These skills offer TTS capabilities for languages not native to the Nao robot. The Furhat Remote API, utilized for intensive speech and listening exercises with children who stutter, can assist and facilitate the therapist to mimic voluntary stuttering by enabling the robot to repeat, block or prolong sounds. We present below some specific observations and details related to managing stuttering in various languages. According to American Speech-Language-Hearing Association (ASHA) (https://www.asha.org/public/speech/disorders/stuttering/) various forms of disfluencies are observed in stuttering, including the repetition of parts of words (repetition: <en>“mum mum mum my name is”;<bg> “ЗЗЗЗдравей“ ), have a hard time getting a word out (blocks: <en>”I lullll love to play”;<bg>”здра а а вей”) or stretch a sound out for a long time (prolongations-<en>“with my fufuffff family“;<bg> “здраааавей” ). The code snippet in Figure 2 demonstrates how Furhat can perform voluntary stuttering therapy. The text provided for simulating the robot's stuttering may vary depending on the language. It is easier in languages where letters correspond to specific sounds. This is not the case in English, where the presence of spaces and additional sounds, such as 'u' is required for Furhat to pronounce the sound, rather than just the letter.

```
from furhat_remote_api import FurhatRemoteAPI
furhat = FurhatRemoteAPI("Furhat_IP_address")
furhat.set_voice(name=<voice>)
# Disfluencies during stuttering: repetition of parts of words
furhat.say(text="mum mum mum my name is Furhat", blocking=True)
# hard time getting a word out (blocks)
furhat.say(text="I lullll love to play ", blocking=True)
#prolongation a sound
    furhat.say(text="with my faaaaamily", lipsync=True, blocking=True)
```

*Fig. 2. Python code snippet: voluntary stuttering with Furhat remote APIs*

The experimental findings confirmed the practicality of the software architecture and also highlighted its effectiveness in facilitating genuine and natural conversations with and among the robots. Additionally, the outcomes clearly illustrated the software's ability to promote smooth collaboration among the robots, offering a strong indication of its practical usefulness in real-world situations. The positive outcomes of these trials not only reinforce confidence in the software's potential to enhance interactions between humans and robots but also create opportunities for ongoing exploration and improvement. This propels the field of conversational robotics toward more advanced and refined capabilities. In the future, we will design and explore other therapeutic techniques that require Furhat's cooperating by utilizing its Dialog Framework. These techniques will include turn-taking, where the robot initiates the dialogue with varying pauses (e.g., Robot - One, Child - Two, Robot - Three). Moreover, the Furhat Dialog Framework can contribute to the development of morphological skills, encompassing gender and number agreement, using separate cards for numbers (1, 2, 3) and different genders (e.g., strawberry, dog, horse). Additionally, blending and segmenting, which are fundamental phonemic awareness skills crucial for reading and spelling, will be tested.

## 4. CONCLUSION

In conclusion, the presented software architecture not only offers a valuable resource for developers but also serves as a pivotal tool in advancing the conversational capabilities of robots. By tapping into NLP Cloud services tailored to specific use cases, developers can harness a versatile and adaptable framework. The architecture adopts a flow-based approach, incorporating a shared repository and message-based input/output channels. These features facilitate the seamless exchange of natural language transcriptions among robots, fostering a collaborative environment conducive to cooperation. The validation of this software architecture has been meticulous, with a particular emphasis on functionalities that highlight collaboration among Pepper, NAO, and Furhat robots. This focus, especially in the realm of speech and language therapy, demonstrates the practical applicability and effectiveness of the proposed architecture. As technology continues to evolve, this innovative approach not only showcases the potential for enhanced human-robot cooperation but also paves the way for future developments in the realm of conversational robotics and therapy applications. The robust validation process underscores the reliability and efficiency of the software architecture in real-world scenarios, affirming its relevance and utility for the broader robotics community.

## ACKNOWLEDGMENT

This research was supported by the National Scientific Research Fund, Project № КП-06-Н67/1.

## REFERENCES

[1] Małgorzata Gawlik-Kobylińska. An approach to the design of formal and informal education services for the system architecture: The lens of organization, competence, pedagogy, and technology. *International Journal on Information Technologies and Security*, vol.15, no.3, 2023, pp. 107-114. https://doi.org/10.59035/EYEH2143

[2] A. Lekova, P. Tsvetkova and A. Andreeva, "System Software Architecture for Enhancing Human-robot Interaction by Conversational AI," *2023 International Conference on Information Technologies (InfoTech)*, Varna, Bulgaria, 2023, pp. 1-6, doi: 10.1109/InfoTech58664.2023.10266870.

[3] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, and F. Tanaka. Social robots for education: A review. *Science Robotics*, vol. 3, no. 21, p. eaat5954, Aug. 2018, doi: https://doi.org/10.1126/scirobotics.aat5954.

[4] A. O. Elfaki et al. Revolutionizing Social Robotics: A Cloud-Based Framework for Enhancing the Intelligence and Autonomy of Social Robots. *Robotics*, vol. 12, no. 2, p. 48, Apr. 2023, doi: https://doi.org/10.3390/robotics12020048.

[5] F. Kaptein et al. A Cloud-based Robot System for Long-term Interaction: Principles, Implementation, Lessons Learned. *ACM Transactions on Human-Robot Interaction*, vol. 11, no. 1, pp. 1–27, Mar. 2022, doi: https://doi.org/10.1145/3481585.

[6] C. Deuerlein, M. Langer, J. Seßner, P. Heß, and J. Franke. Human-robot-interaction using cloud-based speech recognition systems. *Procedia CIRP*, vol. 97, pp. 130–135, 2021, doi: https://doi.org/10.1016/j.procir.2020.05.214.

[7] Grassi, L., Recchiuto, C.T. & Sgorbissa, A. Sustainable cloud services for verbal interaction with embodied agents. *Intel Serv Robotics*, 16, 2023, pp.599–618 https://doi.org/10.1007/s11370-023-00485-3

[8] M. Elgarf, G. Skantze, and C. Peters, "Once Upon a Story. KTH Publication Database DiVA (KTH Royal Institute of Technology), September 2021, doi: https://doi.org/10.1145/3472306.3478359.

[9] B. Irfan, Sanna-Mari Kuoppamäki, and G. Skantze. Between Reality and Delusion: Challenges of Applying Large Language Models to Companion Robots for Open-Domain Dialogues with Older Adults. *Research Square (Research Square)*, May 2023, doi: https://doi.org/10.21203/rs.3.rs-2884789/v1.

[10] Fan J, Mion LC, Beuscher L, Ullal A, Newhouse PA, Sarkar N. SAR-Connect: A Socially Assistive Robotic System to Support Activity and Social Engagement of Older Adults. *IEEE Trans Robot*. vol. 38, no.2, Apr 2022, pp. 250-1269. doi: 10.1109/tro.2021.3092162. Epub 2021 Aug 5. PMID: 36204285; PMCID: PMC9531900.

## *Information about the authors:*

**Anna Lekova** is a professor and head of "Interactive Robotics" Department in the Institute of Robotics-BAS. She has over 450 citations and over 150 publications with IF or SJR. Her current research interests are in the Interactive Robotics, SARs, Fuzzy Logic, BCI and IoT.

**Paulina Tsvetkova** is an assistant professor at the Institute of Robotics –BAS. She has a PhD in Psychology and certificates in Python programming. Currently, she is doing a course in Artificial Intelligence. Her publications are in the field of Emotional intelligence, Vocational interests and Robot-assisted psychosocial techniques for children with hearing impairments.

**Anna Andreeva** is an assistant professor in Logopedics at South-West University "Neofit Rilski" and a researcher in the Institute of Robotics-BAS. She has scientific interests in the field of hearing and voice disorders, augmentative and alternative communication, assistive and robotic technologies applied in speech and language therapy with people with communication disorders.

**Miglena Simonska** is an associated professor in Logopedics and dean of the Faculty of Public Health, Health Care and Sport at South-West University "Neofit Rilski". Her scientific interests are in the field of stuttering and neurologically based communication disorders in children and adults.

**Adelina Kremenska** is a PhD student in the Institute of Robotics-BAS. Her current research interests are in the Brain-Robot Interactions, Flow-based Programming and IoT.

**Manuscript received on 29 November 2023**