Workshops at 18th International Conference on Intelligent Environments (IE2022)
H.H. Alvarez Valera and M. Luštrek (Eds.)

doi:10.3233/AISE220018

An Architecture for Social-Aware Navigation Based on a Chatbot Interaction

Érika CORREIA [a,1], Anderson LEITE [a], Giovane FERNANDES [a], João VILASBOAS [a], Marcelo SAMPAIO [a], Adriel BASTOS [a], José DÍAZ-AMADO [a,b], João SOARES [a], Yudith CARDINALE [b,c]

[a] *Electrical Engineering, Instituto Federal da Bahia, Vitoria da Conquista, Brazil*
[b] *Electrical and Electronics Engineering Department, Universidad Católica San Pablo, Arequipa, Peru*
[c] *Universidad Internacional de Valencia, Spain*

Abstract.
The coexistence of service robots in social environments has been intensified in recent years, demanding Human-Robot Interaction (HRI) increasingly fluid and necessary. In this context, the present work aims to develop an architecture, called Erika. Erika provides a chatbot to interact by voice and text commands with a service robot, who implements an autonomous navigation, respecting social restrictions based on proxemic zones. An API is available in Erika for connecting the chatbot with a website application, that in turn establishes communication with the Robot Operating System (ROS) to perform simulated experiments and test the functionality of the chatbot and the social-aware navigation of the robot. Results demonstrate that the service robot can respond to the commands provided through the chatbot and , so that, finally, it can drive autonomous or manual navigation, when necessary.

Keywords.
Chatbot, API, Navigation, Interaction, Service robots, Social robots.

1. Introduction

Facing the technological context in the 21^{st} century, the human machine interactions take place in popular living. Therewith, the use of computational systems increases the achievement of tasks in most departments in human work, as well as in daily tasks in their home. This way, the research around Human Robot Interaction (HRI) allows great scientific progress in automation, mainly through the integration of service robots in social environments [1, 2].

The smart architecture implementation added to the current machine learning techniques produce a reality more interactive to the human beings, with greater exploration of data, resources, and comfort [3]. We can exemplify this new trend with the creation of means of autonomous conversation incorporated to the attendance in hospitals, hotels, airports, etc, named chatbot to service tasks [4].

[1]Corresponding Author; E-mail:erika.brito.correia@gmail.com

Many recent studies affirm that Artificial Intelligence is increasing the daily digital life through sophisticated software and hardware [5, 6]. In this context, the chatbots provide lots of benefits to their users like easy communication, diversity of languages, and the no need for technical knowledge to understand their operation. The technology to a voice/text interaction system creation may be structured by some moans of processing, such as Natural Language Processing (NLP) [6], a technique of analysis that understands the human language. This technological understanding adds benefits to other areas of social robot, as it can help in navigation through conversation data obtained, then there is the possibility of providing an interaction about the robot movement and its goal [7].

In this context, this work aims to develop an architecture, called Erika, that allows users to interact with a chatbot by voice and text commands with a service robot, who implements an autonomous navigation, respecting social restrictions based on proxemic zones. The chatbot is implemented to communicate with an interactive web page able to detect the user's questions by voice command and to return answers and demand the autonomous navigation of the robot [8]. The robot's functional structure is on the middleware Robot Operating System (ROS) [9]. Aiming this integration, to communicate the chatbot with the website, the architecture proposed provides an API (Application Programming Interface) [10] for connecting the chatbot with the website application, that in turn establishes communication with ROS to perform simulated experiments and test the functionality of the chatbot and the social-aware navigation of the robot.

The main objective in this work is to describe the interactive architecture between the user and a service robot through the API that allows the interaction of a chatbot and the social navigation developed to this kind of robot. The study case refers to a chatbot used to help people with motor disability to move with an autonomous wheelchair. After an analysis about related work in Section 2, the methodology the architecture is presented in Section 3. Results are presented in Section 4. In Section 5, the final considerations are presented.

2. Related Work

The most common approach described in the literature to communicate different modules in a system is the use of external API. There are some API available for free to improve the performance of chatbots and establish the communication with different platforms, by supporting instant message platforms (Line, Twitter, Facebook and Telegram) [11], in this way, hosting the virtual service robot inside them.

In [12], it is presented AGATA, a chatbot for the dissemination of environmental education, whose development is based on existing API to answer the needs of the project. As an example of its suitability, AGATA server was integrated with Telegram, because the platform offers this connection [13]. Another study with the same approach is presented in [14], a chatbot to academic history monitoring in Universities. The system was constructed to answer requests of academic information to the students' parents. The system involves many process, such as how to use the webhook [15] to receive messages and use the Bot API available by Telegram to send messages in answer.

In the context of robotics, the greatest efforts are focused in trying to make a robot the most similar with a human. In such as case, the dialog flow plays a fundamental role in the modeling of a chatbot, in which the use of API is an important aspect to consider [16]. In [17], a chatbot developed with the Dialogflow platform is presented to interact with a social robot with autonomous navigation and able to talk and scan the corporal temperature of people to help in COVID-19 triage.

The social robotics navigation in environments crowded by humans is an area that starts to consider the proxemics theory proposed by Hall [18] and the five proxemic dimensions [19]: Distance, Identity, Localization, Movement and Orientation (DILMO). Thus, social robots adopt social restrictions in their navigation, respecting the rules of proxemic, having as main interest to avoid reaching in intimate or personal spaces of humans when moves through the environment [20]. In [21], it is described a system called GProxemic Navigation able to complement the autonomous navigation, respecting the individual space of each person according to the environment, applying the concept of proxemic zones in robotics, facilitating the insertion of service robots in a social environment. GProxemic navigation system is able to obtain the geographic coordinates as well as regional features of the place where the robot is inserted. This information is processed to determinate the proxemic zone, performing the autonomous navigation with the social momentum algorithm. In this way, based on the robot's location it is possible to determine the distance that it must respect in its navigation.

Robot programming involves a lot of complexity to develop software, thus tools to simulate the robot programming are available. ROS is composed of libraries that make possible to accelerate the development and to decrease the software complexity in robotics. ROS provides a distributive architecture that allows the sharing of information without depending on a link only. Rosbridge is a web-socket serve that establish an extra layer for ROS. This bridge allows non-ROS applications to communicate with ROS applications. Roslibjs is a library of ROS based in Javascript that is used in particularly applications using IP sockets, authorizing them to do the interface with ROS apps [22].

3. Erika: Social-aware navigation architecture based on a chatbot

This section describes the architecture developed and the implementation of the communication process between the Web Application and the Chatbot, the components of Erika system. To develop a study case, a motorized wheelchair (service robot) simulated in the ROS-Gazebo was used. This service robot interacts with the chatbot and performs autonomous social navigation.

3.1. Web Application

The main objective of the developed website is to control the wheelchair simulation in the ROS-Gazebo. The movement is done through interactive arrow-shaped buttons that can be pressed by the user, allowing the wheelchair to move manually in the goal direction, as shown in Figure 1.

This development was based in a web application localized in Github and developed by Robot Web Tools [23] and being composed by the following technologies:

- HTML (HyperText Markup Language) is a markup language that provided the structure of the web page document.

- CSS (Cascading Style Sheets) which is a style language used to stylize the page elements.

- JavaScript, a programming language that provided the page event manipulation, as example the clicks in buttons and the integration with the roslibjs. This language made

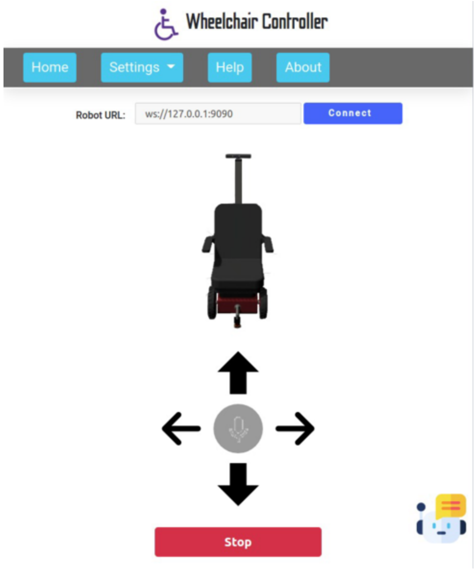

Figure 1. Wheelchair Controller

possible the API Fetch use, which has the fetch global method used to search for network information asynchronously, consuming just the API data.

- Bootstrap is a framework used to develop front-end, in our work it was used to make the buttons in the page

- Roslibjs [24] is a library JavaScript which allows the interaction with ROS from browser. It uses the WebSockets to connect itself to rosbridge [25], that provide the JSON API functionality [26] assisting ROS and not ROS programs.

3.2. Erika

The Erika is a functional and interactive architecture that aims to facilitate the communication between the user and the service robot by voice and text commands. In the next sections it is presented the architecture, the conversation flow, and the study case in the Erika system.

3.2.1. Architecture

The Erika's architecture is based on a client-server model as shown in Figure 2. The client side consists in a graph interface where the user may interact with the application. After the client command gets in, the chatbot processes the information and gives back the command to run asking the user to press the green button in case of the correct command, in case of incorrect command, the chatbot asks the user to press the red button to cancel.

The server side is divided in two parts, the user by chatbot and the ROS one. The chatbot server interacts with an API according it receives the information, processes it and send to the API. As contribution of the article, it was developed an API used to supply the need of integration between Dialogflow and the web application, which one was developed using Node.js, a runtime JavaScript built in JavaScript V8 from Chrome [27]

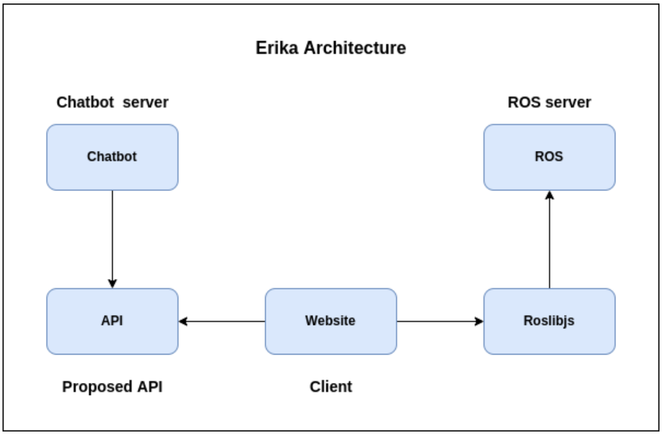

Figure 2. Erika architecture system

together with the Express.js framework highly configurable. One of the main reasons to the Express.js' utilization was its components called middleware that allows the convention configuration [28]. Besides the flexibility provided and high customization in the web application development [28] and provide a set of fundamental resources that helped in the API development, the API is responsible for check the user command existence in a database in JSON format which has a set of data that can be run and originally is defined as false. the user's request action being found in this database, the API update the value to true. To the ROS side works, it is necessary that the user confirm the information returned by the chatbot pressing the green button in web page. From this action, the web page make a request in API searching for a command set as true and, after that, processes and sends to a topic in ROS through roslibjs library. Arriving in ROS, the action chosen by the user will be run and then, the web page make a request to API to update the information before set as true to false.

3.2.2. Conversation Flow

To the beginning of the conversation with chatbot, it's necessary that the chatbot access the web application and after that, select the chatbot icon located in the page. Then, the person can interact with chatbot through text or voice command as shown in Figure 3.

The implementation of the chatbot messages flow is due to the Dialogflow, a Google platform that allows the intents construction, that is the intentions of the conversational agent, where each intent permit adding training sentences and possible answers. In this chatbot, it was created 9 intents, which ones 2 are basic conversation flow like greeting and goodbye, 4 are easy commands to go forward, go back, turn left and right. In the end, it was created 3 intentions to move the autonomous service robot in the simulated environment ROS-Gazebo with the proxemic navigation [21]. That intentions are "reception", "commercial room", and "library". Besides the intents, the Dialogflow has as tools the Agent, the Context, the Event, the Training Phrase, Action and Parameters, Answer, Integrations and Fulfillment that makes possible the conversation accepting in the dialog between chatbot and the user [4] allowing the exchanging of information with the API. With Dialogflow, the use of the chatbot becomes viable and important, because it allows

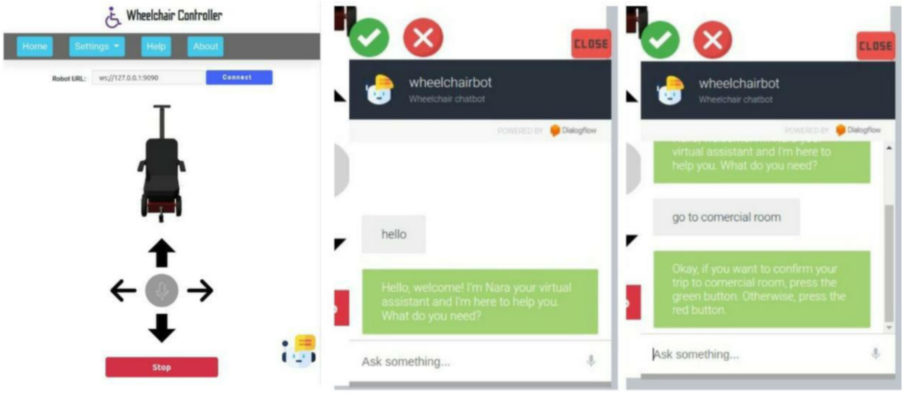

Figure 3. Conversation flow between user and chatbot

agile autonomous navigation without the repetitive need to press existing buttons on the web page.

3.3. Study Case

To improve the project, it was made an autonomous wheelchair simulation in an office at the ROS and Gazebo formed by 3 areas, considering the reception as the environment 1, the environment 2 is the commercial room and finally the library as environment 3. Figure 4 shows the office and its division.

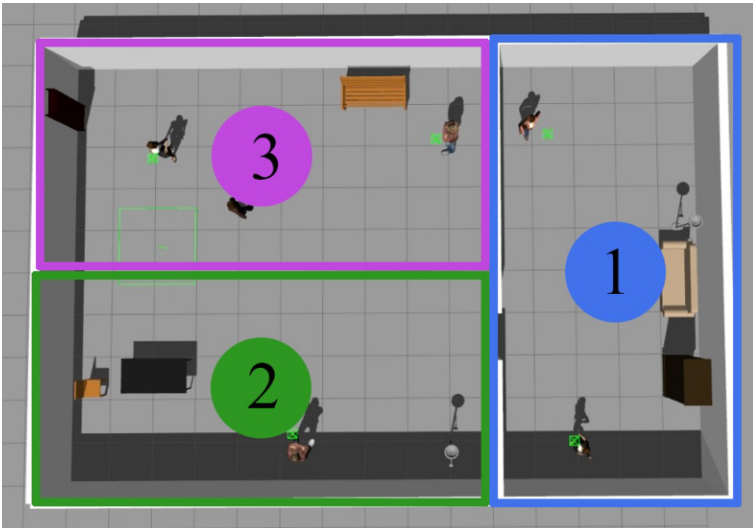

Figure 4. Office and respective area

From that, the first hypothesis can be established, assuming that the user was in the library and wanted to move to the commercial room using the wheelchair, it should start the dialog flow with the chatbot. The user starting the dialog with a greeting will get the chatbot lengths in response, a brief presentation and about its function will therefore ask the user how it can help him. If the user starts the conversation already with a command to activate the robot's movement, it will understand and execute the continuous flow of the conversation. Thereby, when informing that you would like to go to commercial room, at this instant the chat bot will return "OK", if you want to confirm, press the green button. As can be seen in Figure 5.

Figure 5. Interaction between user and chatbot

After user confirmation will initialize the displacement of the wheelchair to the requested environment. Tor interact with the social navigation in the wheelchair, we use the GProxemic system [21], respecting the social restrictions of the environment imposed by this system, so if there are individuals on the trajectory the proxemic navigation is applied to dodge them. In the second hypothesis, with the arrival at the destination if the user decides to approach a table in the commercial room, for this he informs Erika a new command, that this time is to move a little "forward". Again it responds confirming the command and asking the user to press the green button. Making that flow one more time, he arrives at the table, but realizes that he will have to turn right in order to position himself in front of it. Again he asks the chatbot to go to the "right" and thus he manages to reach the your final objective. The direction commands can also be executed using the buttons contained in the web page, adding one more possibility to the user.

4. Implementation Details and Functional Results

The results were demonstrated on two different platforms to prove the implementation of the Erika architecture in simulation environments. The Gazebo and RViz [21] was used to clarify the real trajectory performed by the wheelchair around the map, and Matlab [21] to demonstrate the trajectory of the wheelchair to be done theoretically.

For better understanding, let's review the case study previously presented. The user activate by voice command the chatbot indicating that you want to go to the commercial room, the flow of information cited in Section 3.2.1 occur and this information is published in a topic on ROS as seen in the Figure 6. The wheelchair navigates to the area 2 considering the social restrictions according is illustrated in Figure 7 using GProxemic that helps the displacement by sending a semantic annotation to the wheelchair, since it is inserted in an office where people are walking around, theoretically in Matlab and in Figure 8 this movement can be observed and practically in Gazebo and RViz.

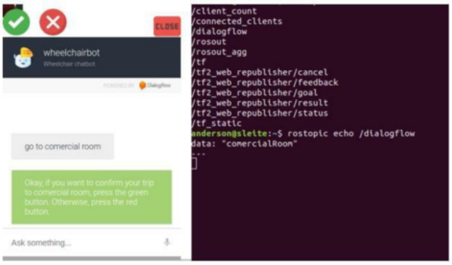

Figure 6. Communication between Architecture Erika and ROS

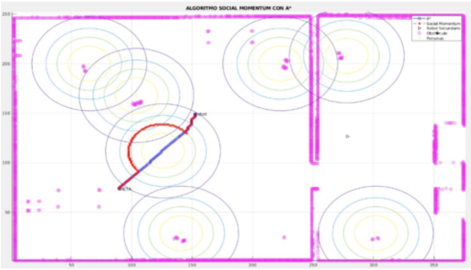

Figure 7. Theoretical wheelchair trajectory in Matlab respecting social restrictions

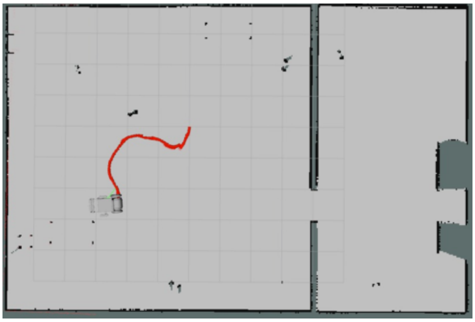

Figure 8. Wheelchair trajectory in a practical way in the RViz respecting social restrictions

The figures presented previously demonstrates that the objective of the proposal was achieved. The Erika's architecture was able to establish communication between the wheelchair and use it efficiently, sending the device to the final point of the trajectory and being activated by voice or text command. While GProxemic send the geographic point so that the device recognizes the environment in which it is inserted and can perform navigation.

5. Conclusion

It was observed from the results obtained that the chatbot is interacting satisfactorily with the user. From this it can be inferred that the trained words are being understood correctly and the information is being transmitted to the ROS-Gazebo allowing the wheelchair to move. Even in cases where people are present, the wheelchair manages to respect their social space and make deviations that take the user to the desired destination. In this way, it was proved that Erika Architecture fulfilled the desired objectives, enabling the exchange of information between the chatbot, the web application and ROS through the API proposed in this article.

In future works may have improvements in the Erika Architecture, making the conversation process with the chatbot even more natural, since the number of words and functions implemented so far is limited to nine. As well as carry out this process through a mobile application, increasing in this way the mechanisms of access to the chatbot implemented in the architecture.

To improve the proposed architecture, it is possible to correct any noise in the communication between user and chatbot that may exist, and also to evaluate strategies that aim to optimize the performance of the Erika Architecture.

Acknowledgement

This research was funded by FONDO NACIONAL DE DESARROLLO CIENTÍFICO, TECNOLÓGICO Y DE INNOVACIÓN TECNOLÓGICA - FONDECYT as executing entity of CONCYTEC under grant agreement no. 01-2019-FONDECYT-BM-INC.INV in the project RUTAS: Robots for Urban Tourism Centers, Autonomous and Semantic-based.

References

[1] Thomas B Sheridan. Human–robot interaction: status and challenges. *Human factors*, 58(4):525–532, 2016.

[2] Rolf Dieter Schraft and Gernot Schmierer. *Service robots*. CRC Press, 2000.

[3] H Wang, Z Lei, X Zhang, B Zhou, and J Peng. Machine learning basics. *Deep learning*, pages 98–164, 2016.

[4] Nudtaporn Rosruen and Taweesak Samanchuen. Chatbot utilization for medical consultant system. In *2018 3rd technology innovation management and engineering science international conference (TIMES-iCON)*, pages 1–5. IEEE, 2018.

[5] Eleni Adamopoulou and Lefteris Moussiades. An overview of chatbot technology. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 373–383. Springer, 2020.

[6] Tarun Lalwani, Shashank Bhalotia, Ashish Pal, Vasundhara Rathod, and Shreya Bisen. Implementation of a chatbot system using ai and nlp. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Volume-6, Issue-3*, 2018.

[7] Jörg Baus, Antonio Krüger, and Wolfgang Wahlster. A resource-adaptive mobile navigation system. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 15–22, 2002.

[8] Xiaoguang Qi and Brian D Davison. Web page classification: Features and algorithms. *ACM computing surveys (CSUR)*, 41(2):1–31, 2009.

[9] Anis Koubâa et al. *Robot Operating System (ROS).*, volume 1. Springer, 2017.

[10] Joshua Ofoeda, Richard Boateng, and John Effah. Application programming interface (api) research: A review of the past to inform the future. *International Journal of Enterprise Information Systems (IJEIS)*, 15(3):76–95, 2019.

[11] Mohd Najwadi Yusoff, Ali Dehghantanha, and Ramlan Mahmod. Forensic investigation of social media and instant messaging services in firefox os: Facebook, twitter, google+, telegram, openwapp, and line as case studies. In *Contemporary digital forensic investigations of cloud and mobile applications*, pages 41–62. Elsevier, 2017.

[12] Bruno Rocha Gomes, Antonio Fernando Lavareda Jacob Jr, Ivan de Jesus Pereira Pinto, and Sérgio Colcher. Ágata: um chatbot para difusão de práticas para educação ambiental. In *Anais Estendidos do XXVI Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 85–89. SBC, 2020.

[13] Nicolas Modrzyk. *Building telegram bots: develop bots in 12 programming languages using the telegram bot API.* Apress, 2018.

[14] A Heryandi. Developing chatbot for academic record monitoring in higher education institution. In *IOP Conference Series: Materials Science and Engineering*, volume 879, page 012049. IOP Publishing, 2020.

[15] Hari Setiaji and Irving V Paputungan. Design of telegram bots for campus information sharing. In *IOP Conference Series: Materials Science and Engineering*, volume 325, page 012005. IOP Publishing, 2018.

[16] M Lombardi, F Pascale, and D Santaniello. An application for cultural heritage using a chatbot. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–5. IEEE, 2019.

[17] Ryan H Kim and Hyung Gi Min. Development of a conversing, temperature scanning, autonomously navigating robot to help screen for covid-19. In *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, pages 980–987. IEEE, 2021.

[18] William B Hart and DOI JaLC. Edward t. hall and the history of intercultural communication: the united states and japan sub title author rogers, everett m.(miike, yoshitaka). 2002.

[19] Saul Greenberg, Nicolai Marquardt, Till Ballendat, Rob Diaz-Marino, and Miaosen Wang. Proxemic interactions: the new ubicomp? *interactions*, 18(1):42–50, 2011.

[20] Marcos Daza, Dennis Barrios-Aranibar, José Diaz-Amado, Yudith Cardinale, and João Vilasboas. An approach of social navigation based on proxemics for crowded environments of humans and robots. *Micromachines*, 12(2):193, 2021.

[21] João Pedro Vilasboas, Marcelo Sá Coqueiro Sampaio, Giovane Fernandes Moreira, Adriel Bastos Souza, Jose Diaz-Amado, Dennis Barrios-Aranibar, Yudith Cardinale, and Joao Erivando Soares. Application of social constraints for dynamic navigation considering semantic annotations on geo-referenced maps. In *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–7. IEEE, 2021.

[22] Saran Khaliq, Shahzeb Ahsan, and M Danish Nisar. Multi-platform hardware in the loop (hil) simulation for decentralized swarm communication using ros and gazebo. In *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 310–315. IEEE, 2021.

[23] Russell Toris, Julius Kammerl, David V Lu, Jihoon Lee, Odest Chadwicke Jenkins, Sarah Osentoski, Mitchell Wills, and Sonia Chernova. Robot web tools: Efficient messaging for cloud robotics. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4530–4537. IEEE, 2015.

[24] Mustafa Karaca and Ugur Yayan. Ros based visual programming tool for mobile robot education and applications. *arXiv preprint arXiv:2011.13706*, 2020.

[25] Christopher Crick, Graylin Jay, Sarah Osentoski, Benjamin Pitzer, and Odest Chadwicke Jenkins. Rosbridge: Ros for non-ros users. In *Robotics Research*, pages 493–504. Springer, 2017.

[26] Lindsay Bassett. *Introduction to JavaScript object notation: a to-the-point guide to JSON.* " O'Reilly Media, Inc.", 2015.

[27] Daejun Park, Andrei Stefănescu, and Grigore Roşu. Kjs: A complete formal semantics of javascript. In *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 346–356, 2015.

[28] Azat Mardan. Using express. js to create node. js web apps. In *Practical Node. js*, pages 51–87. Springer, 2018.