Social Educational Robotics Application: Architecture and Interconnectivity

K. Papadopoulos, A.M. Velentza, P. Christodoulou, N. Fachantidis

School of Educational & Social Policy, University of Macedonia, Thessaloniki, Greece,Laboratory of Informatics and Robotics Applications in Education and Society (LIRES), University of Macedonia, Greece

Abstract:
Social robots are widely used in educational activities. Apart from their hardware features that makes them capable of interacting with students, teachers, and educational stakeholders, the software applications are of great importance for a successful use. In the current paper, we present the design and architecture of the STIMEY robot application. The application was designed to transmit all information to and from the robot, ensuring communication between the robot agent and a) the educational platform, b) other social robots, c) the current user.

SECTION I.

Introduction

Educational robots are widely used to support children's education. Their use is starting from kindergarten and pre-school ages to enhance constructive learning via storytelling techniques[1], computational thinking[2], and STEM principles[3]. The use of educational robots increases as children get older and researchers and educators report beneficial outputs from their use in elementary school students learning outcomes[4], and motivation[5].

One major aspect in the robots' acceptance in school classrooms is their social characteristics[6]. User-centered and participatory design studies, tend to agree that students prefer to interact with social robots which can talk, show emotional expressions with facial and body gestures, give them feedback, and support their educational activities[7], [8]. Similar outcomes have been found when future teachers have been asked to evaluate the ideal characteristics of a robot in order to collaborate with it during educational activities[9]. Those abilities increase the affection between the students and the robot[10].

In the current study, we present the software implementation of the STIMEY Socially Assistive Robotic Artefact (SARA). STIMEY[11] aims to become an educational assistant and companion by increasing students' motivation and rewarding their learning process towards STEM studies. The robotic hardware artifact is interconnected with an Android OS smartphone, representing the “brain” of the robot, as depicted in Fig.1. The use of the Android OS as an educational tool provides access to rich content available to the teachers, such as real-time retrieval of online information, classroom support framework activities, screen-sharing among robots and education material via google play[12].

The robot designed to mimic social behaviors and have anthropomorphic characteristics, head, arms, and face placed on a screen display. It moves backwards and forwards, express emotion with body gestures and facial expressions, as shown in Figure 2. All the social characteristics of the robot decided through a participatory design study, which allowed not only for the identification of the stakeholders requirements, but also for experts analysis that safeguards the design[13].

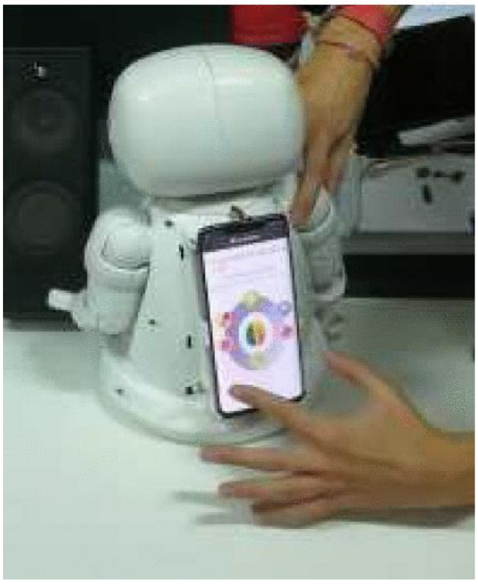

FIG 1. Software And Hardware Interconnectivity With Android Serving As The Robot’s ‘Brain’.

FIG 2. Stimey Sar Appearance And Anthropomorphic Cues

SECTION II.

Related Work

The use of smartphones, serving as the ‘brain’ of educational robots have been proposed due to the easy, flexible and intuitive user interface and programming potentials[14]. Especially the use of Android system offers interconnectivity with robots by wireless communication, by simplifying the robot’s controls[15].

To support robots’ social behavior, it is crucial to combine software and hardware artefacts [16]. Control in educational activities needs to be easily accessible from the users since the educational robots are rarely deployed autonomously for more than a few days[17]. Finally, the robot’s controller is not only important for its own use but also for the interconnectivity with other robots performing in the same environment[18].

SECTION III.

Application

A. Extracting system requirements

For the design of the STIMEY robot SARA, we followed a Participatory Design Approach (PDA). The PDA concerns the direct involvement of multidisciplinary groups of stakeholders in the co-design of products aiming to improve their effectiveness [19]. Still, during the PDA interdisciplinary teams of experts are engaged to ensure that the product will be in accordance with the state of the art in each field [20]. Our previous research highlighted in the preliminary step of the PDA, stakeholders’ requirements concerning the use, the appearance and the voice of the STIMEY robot [21].

B. System Requirements

The software architecture design and implementation were developed and adapted in parallel with the robot’s hardware application programming interface (API) research. Firstly, we implemented a two-way asynchronous TCP serial connection between the robot’s micro-controller and the Android OS device. Based on research we designed our Text to Speech (TTS) and Automatic Speech Recognition (ASR) implementation[22]. The interface was designed in a way to facilitate the implementation of ASR and TTS libraries[23]. Moreover, a comparative analysis of existing applications determined the best practices for communication between asynchronous web-based systems and real time processing engines. Those procedures followed the integration of TTS libraries for conversation and the extensive testing of the ASR and TTS functionality in a) a controller environment and b) a real-world scenario.

Research on latest technologies and best practices around User Interface (UI) and User Experience (UX) for smart devices lead us to design an unobstructive user interface that allows students of all ages to interact with the robot and also the implementation of user interface related functionality.

We also focused on voice inputs, so that the users can access the application's functions in a near natural language way. Enhancing the robot with voice, enhanced the user experience for students taking it to a more personal, multisensory, humanlike engagement with technology, never experienced in the classroom before.

The interconnectivity between the hardware and software parts of the robot was also a priority. We focused on a single-identity login structure to make it easier for the educators to use it and automatically associate the robot with the user in the platform. The architecture of the educational platform has been analyzed in a previous publication[24]. We also designed a multi-lingual interface, where each user's language can be selected before logging in but can also be changed from a relevant menu.

Moreover, based on use-case studies featured by an interdisciplinary group of pedagogues, a developmental psychologist, educational psychologist, HRI experts, electrical and computer engineers we designed a 'Notes' functionality, where the users are able to keep their own notes in the robot. Note reading is supported through on-screen text and TTS while Note writing is supported through ASR or on-screen keyboard. We followed the same procedure for implementation of Notification functionality. The platform can send notifications to the robot, which can either be displayed or verbally transmitted to the user via a TTS functionality.

C. Testing of the STIMEY SARA behavior on the Mock-up platform through the android application

The testing procedure of the use-cases was implemented through the Mock-up platform, and new adjustments were made on the verbal expressions of the robot regarding the spelling of many words specifically in local languages. For instance, for the robot to spell right the phrase "Hey, guys!" in the Greek language, linguistic modifications had to take place. Instead of spelling the phrase "Éi, paidiá!" (=Έι, παιδιά!"), the phrase was programmed to be spelt in a wrong way, namely "Chéi paidyiá" (=Χέι, παιδγιά!), which nevertheless created an "illusion" that the robot correctly spelled the phonetics.

SECTION IV.

Software Design

Due to the unique requirements for the STIMEY SARA, specialized software had to be designed and written to accommodate all the functionality that the robotic artefact would provide. We decided to use off-the-shelf frameworks and thus, based on the framework implementation needs, we proposed the following requirements to the hardware design team, during manufacturing process of the robots. The robot would require an Android smartphone with a custom firmware to facilitate communication through the serial connection to a NanoPi board, which would act as the hardware controller. This allowed us to bypass the phase of low-level design of the framework stack and we could use the existing Android framework to communicate with robot's firmware, which was written in Python.

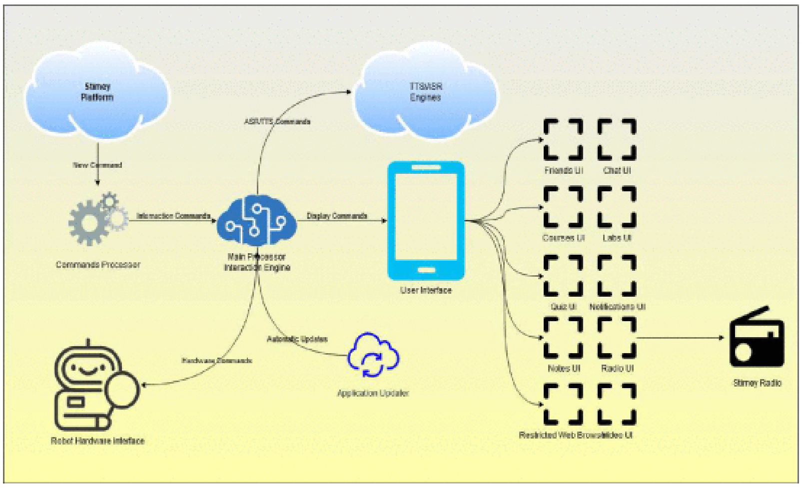

FIG 3. A Software Architecture Design Diagram

In the above diagram (Fig.3) we demonstrate the interconnections between our software stack and the different firmware and engines that reside within the robotic artefact.

• STIMEY Platform

The platform component facilitates all the communication with the STIMEY platform. The platform implements a REST-API, which allows users to authenticate, store and retrieve user data from the platform. The platform also hosts a Message Queuing Telemetry Transport (MQTT) server. MQTT allows for bidirectional communication between clients but also between the server and the client. The software connects to the MQTT server as a client and subscribes using the user's ID for messages directed to the user. The platform knows the user's ID and is able to forward commands and messages through the MQTT pipe, also our software is able to communicate by replying to those messages or communicating directly with the platform and thus facilitate bidirectional communication. When a valid message is received, it is then forwarded to the Commands Processor for analysis.

• Commands Processor

A STIMEY message is a complex structure, which contains multiple sub-messages. A message might contain robot movements, eg. "Raise your left hand", Voice interactions, eg. "Speak: Hello World!", software directives, eg. "Launch: Radio UI", interaction through facial expressions, eg. "Facial: Smile" etc. A custom implementation of a Markdown language was chosen to easily interpret the message. Markdown languages are lightweight formatted text which is similar on how HTML is structured but with a key different being that that is appealing to human readers. For example if someone wrote [b]Hello World![/b] it would output as **Hello World!**. Most people are familiar with those languages from instant messaging or communication forums. The solution would adhere to all the rules set by BBcode Markdown language because it is one of the few markdown languages allowing within sub-structures. A message would arrive as a text, the command would be split in a tiered fashion, allowing the first commands of the first level to be the first that will get executed, any sub-commands would be executed following a tree pattern based on their level and then subsequent first-level commands would be executed afterwards until all the pipeline is exhausted. In the following example message, "[RobotFace type=BlinkLeft] [/RobotFace] [MovementPredefined command=Hello][TTS text=Hello World!] [/TTS][/MovementPredefined]", the robot would first execute the facial command to blink the left eye and then would wave its hand (hardware command) while speaking "Hello World!" (TTS command). Once the message is parsed and all the commands are separated, they are forwarded to the interaction engine, which handles forwarding the relevant commands to each engine and making sure that they are executed in the correct order.

- Interaction Engine

The interaction engine is responsible to execute any commands that are coming up from the command processor along with any interaction that is happening from user's inputs. The interaction engine secure that the commands are executed in correct order and that they are finalized correctly allowing subsequent commands to be executed. The interaction engine also handles any user interaction, i.e. the user touches the head of the robot, the robot is notifying us through the Robot Hardware Interface (RHI) and the interaction engine decides what should do with this interaction. The interaction engine is also acting as safeguards monitoring connections that monitors connections to all subsystems and restarts any connection or subsystem that is not responding within the pre-determined parameters.

- TTS/ASR Engines

The TTS/ASR engines are facilitating logic provided by MLS Inc[1] that allows our software to implement ASR and TTS functionality. The Interaction Engine initializes the TTS/ASR engines and translates any commands or input to functions that are required by the engines. When a TTS command is executed, the text to be spoken is forwarded to the engine, the engine translates the text to audio and then plays it immediately. When the audio is completed the TTS engine notifies through callbacks the interaction engine so it can proceed with the next command. Similarly, the ASR engine when commanded by the Interaction Engine listens for verbal input, transcribes the audio to text and the text is send back to the Interaction Engine to be further processed. Both engines require a stable WiFi connection to be fully functional.

- Robot Hardware Interface

The Robot's Hardware Interface utilizes the smartphone's USB Serial Bus to communicate with the firmware located in the NanoPi. The software will initialize the connection with the firmware on startup and will use the API provided by the hardware team to communicate with the robot's stepper motors, sensors or on-head screen. The robot hardware interface will translate the commands received by the Interaction Engine to binary and then send them over the serial. Once a command is sent, the firmware will wait for confirmation of the command, if a confirmation is not received within the allotted time, it will attempt to reestablish the connection. The hardware did not only reply when it receives the corresponding command. A series of pre-built commands were timed by the Hardware Interface and waited based on default times. Unfortunately, this cannot be implemented for customized commands. Thus, we added a special command on the Commands Processor, the “Wait: 1 sec” command, which allows the teacher to calculate the required ‘wait’ for more complex movement.

• Application Updater

The application updater is an internal process that makes sure that the application is fully up-to-date. The updater will communicate with the back-end servers in regular intervals to check for updates. If an update is found, the updater will download it and forward it to the Android Package Manager to install it without user interaction. This is a special process, which requires our app to be marked as a “System Application”.

• Robot to Robot communication

The robots can communicate with each other using the STIMEY platform and utilizing the MQTT service mentioned earlier. Robots can send commands to other robots using the functionality provided by the Command processor and the Interaction engine. In the following example, a user is discussing with his friend using the Chat functionality, and when a specific scenario is detected (eg. “I wish we could dance now”) both robots can propose to execute a small choreography. The acting robot(s) retrieves the choreography from the platform and broadcast it to the robots connected in the discussion. The robots will process the commands and pass them to the interaction engine, which will make both robots to act on.

• User Interface

The user interface contains all the screens that are displayed to the user. The user interface communicates directly with the Interaction Engine through callbacks in order to pass user input back and forward. The following screens are implemented: *Friends UI:* 1. A friend list is displayed; the friend list is fetched from the STIMEY Platform. 2. Users can click the chat icon to start an interactive chat with their friends. *Chat UI:* 1. A classic chat user interface where two or more users can exchange messages.2. The chat uses the MQTT bidirectional messaging, and no messages are saved to facilitate user privacy. *Courses UI*: A list of subscribed courses is displayed, any linked material with the course is also displayed. This allows a student to find all the material required to complete the course within the robot. *Quiz UI:* Users can undertake interactive quizzes that are specially designed and programmed by the teachers through this specialized interface. *Notifications UI:* Users can receive platform wide notifications or special messages send to them by the STIMEY platform. *Note UI:* Users can keep their notes within the robot using both on-screen keyboard and voice input. *Radio UI:* Users can navigate to the radio UI and listen to the STIMEY radio. User can interactively change between the available radio stations or start/pause their music. *Restricted Web Browser:* This is a specialized interface that can be launched through a command programmed by the teacher. The specialized interface does not allow following any further links to avoid exposure and includes a special filter which is compliant with the Greek School Network Administration. *Video and Image UI:* This is a specialized interface that can be launched through a command programmed by the teacher and will launch a YouTube video directly on the back-screen of the robot. The YouTube will exit as soon as the video is displayed to avoid any unnecessary exposure.

SECTION V.

Discussion and Conclusions

The aim of this study was to present the software architecture of the educational robot STIMEY SARA. The major advantages of its design were the interconnectivity between the software and hardware artefacts that made it possible for the robot to act in a social manner based on movement controls and Text to Speech Software (TTS). ASR was used to control SARA using voice commands. The software can support the interconnectivity of the robot's 'brain', which is an Android OS mobile phone with an educational platform to retrieve educational material. The same platform is easy to use from educators with pre-programmed behaviors to create their own educational materials. Moreover, due to the robot's interconnectivity it is possible to provide quizzes, notifications, chat, and interaction with other robots in the same environment.

Additionally, the teacher can program the robot at an individual level depending on their level of interest and commitment. All teachers can develop educational activities on the platform and can incorporate robot actions that will be used by the student's robot utilizing the robot editor. Finally, all functions of the STIMEY platform are accessible through the mobile phone integrated on the robot.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement N° 709515, 'Science Technology Innovation Mathematics Engineering for the Young'- STIMEY

REFERENCES

[1] M. Fridin, 'Storytelling by a kindergarten social assistive robot: A tool for constructive learning in preschool education', Comput. Educ., vol. 70, pp. 53–64, Jan. 2014, doi: 10.1016/j.compedu.2013.07.043.

[2] E. Roussou and M.Rangoussi,'On the Use of Robotics for the Development of Computational Thinking in Kindergarten: Educational Intervention and Evaluation'Robotics in Educ,Cham,2020,pp.31–44.

[3] D. P. Davison, F. M. Wijnen, J. van der Meij, D. Reidsma, and V. Evers, 'Designing a Social Robot to Support Children's Inquiry Learning: A Contextual Analysis of Children Working Together at School', Int. J. Soc. Robot., vol. 12, no. 4, pp. 883–907, Aug. 2020,

[4] L. P. E. Toh, A. Causo, P.-W. Tzuo, I.-M. Chen, and S. H. Yeo, 'A Review on the Use of Robots in Education and Young Children', J. Educ. Technol. Soc., vol. 19, no. 2, pp. 148–163, 2016.

[5] K. Chin, Z. Hong, and Y. Chen, 'Impact of Using an Educational Robot-Based Learning System on Students' Motivation in Elementary Education'IEEE Trans.Learn.Technol.,vol.7,no.4,pp.333–345,Oct. 2014

[6] A. Henschel, G. Laban, and E. S. Cross, ‘What Makes a Robot Social? A Review of Social Robots from Science Fiction to a Home or Hospital Near You’, Curr. Robot. Rep., vol. 2, no. 1, pp. 9–19, Mar. 2021,
[7] P. Christodoulou, A. A. M. Reid, D. Pnevmatikos, C. R. del Rio, and N. Fachantidis, ‘Students participate and evaluate the design and development of a social robot’ 29th IEEE International Conference onRobot and Human Interactive Communication (RO-MAN), 2020,pp. 739–744.
[8] A.-M. Velentza, S. Ioannidis, N. Georgakopoulou, M. Shidujaman, and N. Fachantidis, ‘Educational Robot European Cross-Cultural Design’, in Human-Computer Interaction. Interaction Techniques and Novel Applications, Cham, 2021, pp. 341–353.
[9] A.-M. Velentza, S. Pliasa, and N. Fachantidis, ‘Future Teachers choose ideal characteristics for robot peer-tutor in real class environment’, vol. 1384, 2020, doi: https://doi.org/10.1007/978-3-030-73988-1_39.
[10] C. Gena, C. Mattutino, G. Perosino, M. Trainito, C. Vaudano, and D. Cellie, ‘Design and Development of a Social, Educational and Affective Robot’, in 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Feb. 2020, pp. 1–8.
[11] ‘STIMEY’. https://www.stimey.eu/home (accessed Apr. 27, 2020).
[12] N. Fachantidis et al., ‘Android OS Mobile Technologies Meets Robotics for Expandable, Exchangeable, Reconfigurable, Educational, STEMEnhancing, Socializing Robot’, in Interactive Mobile Communication Technologies and Learning, Cham, 2018, pp. 487–497.
[13] D. Pnevmatikos, P. Christodoulou, and N. Fachantidis, ‘Designing a Socially Assistive Robot for Education Through a Participatory Design Approach: Pivotal Principles for the Developers | SpringerLink’,Int J Soc Robot.,2021
[14] C. Krofitsch, C. Hinger, M. Merdan, and G. Koppensteiner, ‘Smartphone driven control of robots for education and research’, 2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems, Aug. 2013, pp. 148–154.
[15] X. Lu, W. Liu, H. Wang, and Q. Sun, ‘Robot control design based on smartphone’, in 2013 25th Chinese Control and Decision Conference (CCDC), Feb. 2013, pp. 2820–2823.
[16] J. Diprose, B. MacDonald, J. Hosking, and B. Plimmer, ‘Designing an API at an appropriate abstraction level for programming social robot applications’, J. Vis. Lang. Comput., vol. 39, pp. 22–40, Apr. 2017,
[17] H. Woo, G. K. LeTendre, T. Pham-Shouse, and Y. Xiong, ‘The use of social robots in classrooms: A review of field-based studies’, Educ. Res. Rev., vol. 33, p. 100388, Jun. 2021, doi: 10.1016/j.edurev.2021.100388.
[18] M. A. Hsieh and V. Kumar, ‘Pattern generation with multiple robots’, in Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., Feb. 2006, pp. 2442–2447.
[19] J. Simonsen and T. Robertson, Routledge International Handbook of Participatory Design. Routledge, 2012.
[20] S. Kujala, M. Kauppinen, and S. Rekola, ‘Bridging the gap between user needs and user requirements’, in In Advances in Human-Computer Interaction I, 2021, pp. 45–50.
[21] D. Pnevmatikos, P. Christodoulou, and N. Fachantidis, ‘Designing a socially assistive robot for education through a participatory design approach: Guiding principles for the developers.’, Int. J. Soc. Robot..

[22] S. D. Craig and N. L. Schroeder, ‘Text-to-Speech Software and Learning: Investigating the Relevancy of the Voice Effect’, J. Educ. Comput. Res., vol. 57, no. 6, pp. 1534–1548, Oct. 2019
[23] M. Santos-Perez, E. Gonzalez-Parada, and J. M. Cano-garcia, ‘Mobile embodied conversational agent for task specific applications’, IEEE Trans. Consum. Electron., vol. 59, no. 3, pp. 61o–614, Aug. 2013,
[24] D. Bauer, B. Penz, J. Mäkiö, and M. Assaad, ‘Improvement of an Existing Microservices Architecture for an E-learning Platform in STEM Education’, presented at the The Fourteenth Advanced International Conference on Telecommunications, Jul. 2018.