Implementing ’Gordon’: A NAO Robot Receptionist to Enhance Efficiency in Critical Services Using Choregraphe and Python

Diana Milligan*, Filip Hanuš*, Harry Williams*,
Jack Thompson*, Natalie Leung*
*School of Engineering, College of Art, Technology and Environment,
University of the West of England, Bristol, UK
Email: {diana2.milligan, filip2.hanus, harry4.williams, jack2.ould, wing7.leung}@live.uwe.ac.uk

***Abstract*—This paper presents what is in this abstract.**
***Index Terms*—NAO Robot, Human Robot Interaction.**

I. Introduction

The purpose of this project is to use a Nao robotic platform to effectively perform a task that requires direct interaction with human. There are many different metrics that can be used to quantify the success of such a system, but as a broader definition an effective system should be able to receive and convey relevant information to an untrained user to achieve a wider goal.

A task that lends itself to this project brief is a reception environment, particularly in a medical setting. According to Mallorie [3], managerial staff shortages within the NHS has pushed clinical staff to spend more time on administrative task over patient-facing care. To effectively reduce staffing requirements and thus make the running of medical centres less resource intensive, robotic systems can be integrated to alleviate the bulk of repetitive tasks. A good low-risk opportunity for this kind of integration is within a receptionist role where any possible errors are of a significantly lower severity than other roles (e.g. medical diagnosis and treatment). A study conducted by Sutherland et al. [5] tested the concept of a robotic receptionist for medical purposes, concluding that the robot displayed a ”professional level of friendliness” that made it suited the role. However, the testing used a Wizard-of-Oz method with the robot only used as the interface between a user and a remote operator. Had all the behaviours been generated by the robot, there could have been a significant difference in how it was perceived by users.

Methods to create a sense of authority for robots has been explored through many different ways, many of which can be implemented within this project. Rae et al. [4] explored the effect of height on robotic telepresence systems, concluding that a shorter ”leader” was less persuasive for a the human follower. As well as the robot’s stature, Rossi (need citation) found that contextual information such as the location of an interaction helped users infer the robot’s role.

Appearance has been argued to be a less important factor by Natarajan and Gombolay (need citation) who tested a variety of different variables for artificial agents, with behaviour being ” the most significant factors in predicting the trust and compliance with the robot” rather than physical appearance. A Nao platform has been used by Cormier et al. [2] to test the limitations of such compliance, finding that users would refuse to follow challenging orders from a robot if factors such as perceived intelligence or professionalism were compromised.

II. Scope and Assumptions

In Scope:

- Initiate client/robot interaction with a waving motion
- Greet client and identify the staff member they have an appointment with
- Nao locates the appointment and emails staff that the client has arrived
- Create gestures that Nao performs during client/robot interaction

Out of Scope:

- Exploring alternative robots
- System is not connected to a database of appointments

Assumptions:

- Interactions will converse in British English only.
- Nao will be positioned out of reach from client users
- Nao will always be powered on

III. Distribution of Work

This project has been equally distributed to and completed by the authors of this report.

IV. Literature Review

TBD

V. Design and Methodology

The NAO-based receptionist system was designed with two main objectives:

1) Streamlining patient check-in and
2) Maintaining user trust and engagement.

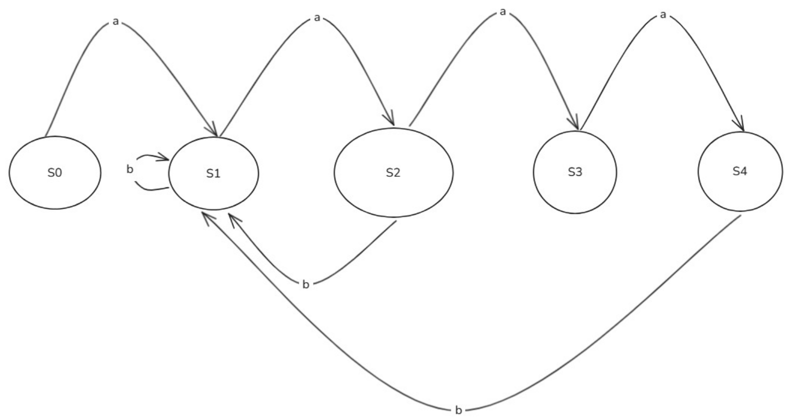

Figure 1. Different States and their Connections

A. Overview of System States

Figure 1 illustrates our finite state machine, implemented using Aldebaran's Choregraphe and Python scripts:

1) **Face Detection (State 0):** NAO scans staff members' faces and stores in database. After completion, state 1 is triggered.
2) **Wave Detection (State 1):** A MediaPipe-based script running in Python continuously scans for a waving hand signal. Upon detection, the system triggers state 2.
3) **Verbal Interaction (State 2):** NAO initiates a spoken dialogue to ask for appointment details. It uses inbuilt speech recognition configured for British English. If the user does not wish to proceed, the system will revert to state 1, ready for the next user. If the user successfully checks in, or they wish to speak to staff, state 3 will be triggered.
4) **Email Notification (State 3):** NAO sends an automated email to the relevant staff member. To circumvent network restrictions on SMTP, local testing used a laptop-based hotspot.
5) **Feedback Collection (State 4):** NAO requests a brief rating from the user, storing any comments for later review. Once completed, the system returns to state 1.

B. Hardware and Software Setup

Choregraphe runs on a laptop that communicates with the NAO robot over Wi-Fi. For gestures, a laptop-mounted camera and a Python-based MediaPipe module detect and interpret waving motions. When network blocks occurred (e.g. institutional firewalls), connections were redirected through a personal hotspot to ensure uninterrupted email functionality.

C. Dialogue Capabilities

Choregraphe boxes were primarily used to facilitate dialogue capabilities. Edited *Say* boxes were added to Choregraphe's *Flow Diagram Panel*, and contained sentences that NAO would say to interact with the user via in-built text-to-speech (TTS) technology. As communication between boxes is event-based, *Say* boxes are triggered onStart and I/O set as "bang". Naturally, NAO asks the user questions to locate their appointment; therefore, *Say* boxes are primarily followed by Speech Reco. boxes to understand the user's verbal response. These use an in-built speech-to-text (STT) technology. Each *Speech Reco.* box contains a word list as part of its parameters, these are now words that NAO recognises, and, by utilising a switch case, can respond accordingly to what the user said. *Speech Reco.* boxes have input set to "bang", output on word recognition set to "string", and confidence threshold was adjusted to 45

A Natural Language Processing (NLP) sentiment detection script was designed but could not be implemented due to integration issues with the NAO robot. This script would have been used for both user inputs and NAO's verbal outputs.

D. Person Tracking

Person tracking capabilities of the NAO receptionist included:

1) Face detection and
2) Wave detection

Face detection was used to detect staff members' faces and stored on the server database. This facilitated NAO's capability of announcing the staff's arrival when they greet the user. Face detection was created using MediaPipe.

Wave detection was implemented to trigger NAO's verbal interaction with the user. A wave was chosen as a common emblematic non-verbal communication gesture and prevents NAO being active by passersby. To prompt the user, a sign was placed in front of NAO saying, 'Wave to speak with the receptionist.' Wave detection used MediaPipe, specifically using the hand landmark and gesture classification models.

E. Email Integration

To meet the requirement of the NAO robot being used as a receptionist, a simple email script was designed to run locally on the NAO hardware using a simple mail transfer protocol (SMTP).

F. Code Overview

During the design process, the code was written in Visual Studio (VS) with version control via GitHub, facilitating unit testing in a controlled environment. Once the code was verified and validated, it was integrated into Choregraphe nodes and further tested on a virtual NAO.

The code was separated across a server and a client. More computationally expensive functions run on the server, whereas simple functions, critical to core functionality, run on NAO. These communicated via a socket framework.

Running gesture recognition and sentiment detection on a server allows deep learning algorithms to benefit from the computing power of the server. This is critical for successful real-time interactions. The server also utilises frameworks that do not integrate well with Python 2 (e.g. MediaPipe and VADER). Additionally, having these features hosted on a server, multiple NAO robots can utilise these tools and could be extremely useful for clients wishing to purchase and use multiple reception robots.

The remaining code, including TTS, STT, and email functions, runs locally on NAO. It uses minimal computing power and can easily be run in real-time on NAO's hardware.

G. GitHub

A shared GitHub housed README files, NAO scripts, and LaTeX documentation for report writing. Branches broke down project tasks and features, whilst version control assisted with project collaboration and testing. See Appendix for supporting GitHub documentation.

H. Design Rationale

We opted for a height-aligned placement of NAO to improve authority and user engagement, as recommended by Rae et al.[4]. This also aligns with Sutherland et al.[5], who showed the benefits of a robotic receptionist in clinical settings. This project expands on their Wizard-of-Oz study by allowing NAO to generate all behaviours without remote operators.

I. Contingency Measures

In practice, user trust decreases sharply when robots show consistent errors, as reported by Bistolfi [1]. Therefore, clear fallback steps were embedded:

- If the appointment check-in fails, staff are alerted via email.

J. Hazard Mitigation

Following a risk assessment, a highlighted safety concern of NAO are the 'pinch points' during robotic movements. To avoid physical contact, NAO will be placed behind a desk and out of reach from the user.

VI. Testing and Results

TBD

VII. HRI Dialogue Example

[User wave detected] Gordon: Hi! My name is Gordon. Do you have an appointment? User: Yes, I do. Gordon: Who do you have an appointment with? User: [unrecognisable mumbling] Gordon: Sorry, didn't catch that. User: I have an appointment with Bob. *[Gordon checks the database for an employee called Bob. Bob doesn't exist.]* Gordon: May I ask for your confirmation number? User: My appointment number is 4. *[Gordon checks the database for a matching confirmation number. Confirmation number exists.]* Gordon: I have got you. *[Gordon sends staff member an email]* Gordon: I have sent an email to the member of staff. Someone will be with you shortly. In the meantime, how would you rate your experience with me on a scale of 1 to 10? User: 10! Your service was fantastic. Gordon: Your feedback is greatly appreciated, thank you. *[Gordon detects the staff member walking in]* Gordon: Our staff is here to help you. Bye.

Scenario: A patient checks in with NAO. The user has an appointment with a staff member, however, that staff member does not exist, NAO prompts the user for their confirmation number. NAO locates the appointment and sends an email to the correct employee. Whilst the user is waiting, NAO asks for user feedback. NAO detects the staff member and announces their arrival.

This scripted scenario highlights two particular failure points anticipated in the system design. Firstly, NAO cannot recognise the user's first response detailing who they have an appointment with. Secondly, the user mistakenly provides the name of someone who does not work there. The system is designed to allow a maximum of two attempts to find the employee, then the user is prompted to provide a confirmation number. If the appointment is located, the script follows as above. Alternatively, if the appointment cannot be found, NAO can follow an expanded script asking if the user would like to speak with a human and emails the manager to assist. NAO can determine these failure points as the Python script contains a predetermined list of the employees. NAO listens for these predetermined words. If the user fails to correctly articulate an employee's name from the predetermined list, NAO will then proceed with the contingencies put in place as part of the system design. To maintain a positive user experience, the user will only be prompted to repeat themselves once, beyond this, human intervention will be offered to avoid user frustrations with the robot.

VIII. Future Work

Facial recognition was planned and developed but was not able to be featured in the final testing due to technical difficulties. Future work on this project would include and expand upon this capability as recognising (and potentially analysing) individuals vastly increases the functionality and perceived intelligence of the system.

To improve the quality of the current conversations, integrating a more robust NLP algorithm increases the likelihood that the Nao robot correctly understands the user's intent from a single output. Currently, any output outside of expected values is considered an error and handled accordingly. NLP allows more unexpected inputs to be parsed for key information that otherwise would've been missed, directing the robot towards a more appropriate response.

While these improvements would improve the technology, extensive user testing (especially within the intended medical space) is the best way to understand the system's shortcomings and improve accordingly.

IX. Discussion

In general, HRI is an element that should be considered early in the design stage, allowing engineers to understand limitations of the system and manage user expectation.

A. Plan vs implementation

Discuss how the final produced HRI solution differs from the initially presented state machine and dialogue. Include a discussion about the reasons for the differences.

Compared to the initial design, there were a few parts added and some not included in the final solution produced. Added elements including gesture detection and NLP for a better user experience. Did not implement integration of NLP, external speech recognition for complex input, and creating face recognition library.

The gesture detection using opencv and mediapipe libraries to detect hand and landmarks, classifying the user's gesture. As co-speech gesture, it provides additional or reiterate information of the conversation to robot. In this application, wave detected indicates conversation starting or ending with users. Waving is an emblem gesture with a defined meaning of "Hello" or "Goodbye", that can help getting attention in human to human conversation, and usually the another person wave back. So it was decided that waving used as a key point of conversation starting in the robot and wave back as well. Making the interaction more realistic.

NLP includes test processing and nomization, tokenization, stemming, and lemmatization, part-of-speech tagging and syntactic parsing. It processes complicated text and get the abstract of idea. Such as analyzing sentiment of user feedback in this case. It was added to better understand complex input and extract useful information, and potential for further improvements based on the user feedback. NLP was implemented with Python code but not integrated with NAO. Because of the complex sentence, it was hard for NAO to recognize and transfer to text. That could be be improved in the future. Speech recognition for complex input was considered as a potential solution fitting with NLP but was not implemented because of the time limitation and subcription required for doing that.

Face recognition library was in the initial design, intended to learn different staff faces, then recognize and introduce them in front of the patient. It was not implemented in the final solution because of technical difficulty using the learn face and face recognize function from Choregraphe. Only one in ten attempts, learning and recognize face success. Different faces were tried and using the face detect first to only start learn or recognize face if there is one in front. Adding delay and loops for testing purpose also did not work. Would be a possible future improvement.

B. Project Limitations

1) **Limited speech recognition.** NAO's speech recognition can struggle with diverse accents and dialects, and recognizing long and complex sentence. Integrating more advanced speech recognition software such as Microsoft Azure or Google Speech-to-Text can improve this with better accuracy. Moreover, NLP from nltk library such as tokenize can be used to understand complex input. Openai can be used to analysis input and improve NAO response to be more intelligent.
2) **Restricted physical interaction.** NAO's limited mobility may offer unrealistic or unidentifiable gestures to its users. To avoid confusions, NAO should maintain basic gestures, and, if required, a video feed on a tablet could supplement NAO's intentions.
3) **Accessibility issues.** Medical environments are subject to users with a diverse range of accessibility needs. The NAO robot required a user to see a sign, prompting the user to wave and initiate communication with the robot. If a user has sight impairments, NAO would not successfully check-in the user. Advanced image recognition could be used to detect visual aids such as guide dogs or mobility canes. Upon detection, NAO could automatically be triggered into state 2, proceeding with checking-in. To build upon the robust nature of NAO in an environment with accessibility needs, a mitigation plan for each ailment could be developed along with supporting improvements to assist these users.
4) **User acceptance and usability.** Older users may be uncomfortable interacting with robots and may find NAO confusing or intimidating. Providing visible instructions and support materials such as posters could encourage familiarity and engagement. Can be improved by doing system test with user like using coded prototype and doing interviews to collect feedback.
5) **Unstable face recognition.** The face learning function which creates the library and recognition of face from the library were unstable. Using external camera like on laptop, and use other algorithm like YOLO for live video face detection and recognition can improve performance. Providing more stable face recognition on identifying staff.

C. Module Feedback

The HRI module was both interesting and highly engaging. It achieved an excellent balance between critical theoretical knowledge on HRI system architectures and practical application using the NAO robot. The module's emphasis on user-centred design and enhancing the overall user experience is an important skill to create successful projects.

A standout highlight was the BRL visit, which offered invaluable exposure to real-world applications of HRI technologies and reinforced the theoretical concepts covered in the module.

One potential improvement would be to increase contact time with the NAO robot, either through a loan system or extended lab sessions. This would allow students to spend more time experimenting with programming and troubleshooting fostering deeper learning and confidence for the assessment.

X. Conclusion

TBD

References

[1] I.M.F. Bistolfi. The influence of anthropomorphism on our feelings for faulty robots, March 2022.

[2] International Conference on Human-Agent Interaction (iHAI). *Would You Do as a Robot Commands? An Obedience Study for Human-Robot Interaction*, 2013.

[3] Saoirse Mallorie. Staff shortages: what's behind the headlines?, 2024.

[4] Irene Rae, Leila Takayama, and Bilge Mutlu. The influence of height in robot-mediated communication. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 1–8, 2013.

[5] Craig J. Sutherland, Byeong Kyu Ahn, Bianca Brown, Jongyoon Lim, Deborah L. Johanson, Elizabeth Broadbent, Bruce A. MacDonald, and Ho Seok Ahn. The doctor will see you now: Could a robot be a medical receptionist? In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4310–4316, 2019.