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. Distribution of Work

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

II. 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.

III. Scope and Assumptions

In Scope:

- Begin client/robot interaction with a waving motion
- Greet client and recognise which staff member they have an appointment with
- NAO finds the appointment and emails staff that the client has arrived
- Create natural gestures that NAO performs during client 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

IV. Technical Implementation

A. State Machine Description

1) System Start:

a) Entry Condition::
The robot/system is powered on or reset. This is the initial state.

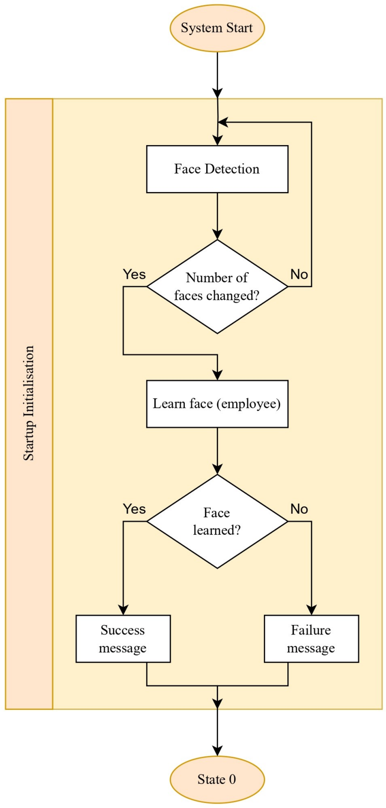

Figure 1. Startup Initialisation

b) Actions::

- Immediately triggers **Face Detection**, looking for any faces in view.
- Checks whether the *number of faces* has changed (i.e., a new face appears).
- If a new face is detected, the system attempts to *Learn face (employee)*. If learning is successful, it proceeds with a *Success message*; if unsuccessful, it gives a *Failure message*.

c) Transitions::

- Both the *success* and *failure* learning branches eventually converge and transition to **State 0**. This is in order to prevent getting stuck due to NAO's technical problems.

2) State 0:

a) Entry Condition:: The machine enters this state after finishing the initial face-detection/learning process.

b) Actions::

- Continuously *reads webcam input*.
- Monitors for a *wave* via MediaPipe.

c) Transitions::

- If a wave is detected, perform *Action: Wave back* and move to **State 1**.
- If no wave is detected, remain in **State 0** and continue monitoring.

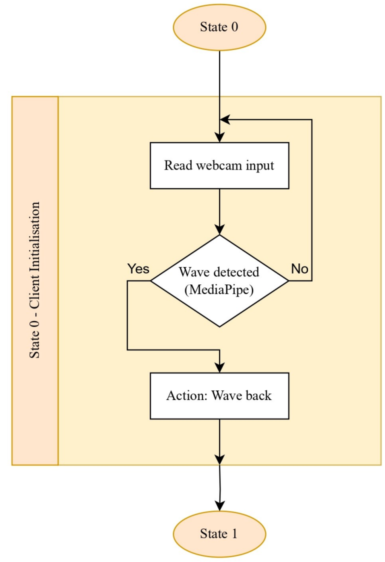

Figure 2. State 0 - Client Initialisation

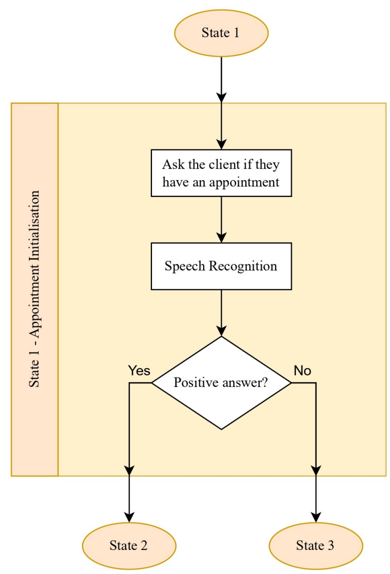

Figure 3. State 1 - Appointment Initialisation

3) State 1:

a) Entry Condition:: A wave has been detected in **State 0**, triggering the move here ("Client Appointment Initialisation").

b) Actions::

- Asks the user if they have an appointment.
- Performs *Speech Recognition* to capture their answer.
- Checks if the answer is "yes" or "no".

c) Transitions::

- If *yes*, transition to **State 2** (handle appointment details).
- If *no*, transition to **State 3** (offer human assistance).

4) State 2:

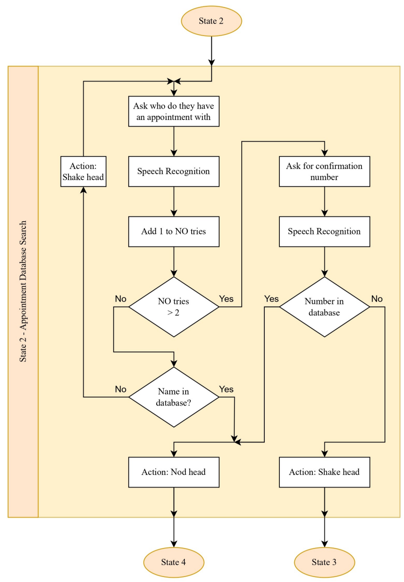

Figure 4. State 2 - Appointment Database Search

a) Entry Condition::
User has indicated they do have an appointment.

b) Actions::

- Prompts for the name of the staff member or the confirmation number.
- Uses *Speech Recognition*, then checks if the provided information is found in the database.
- Keeps track of the number of failed attempts.

c) Transitions::

- If the name or confirmation number is valid, *nod head* and transition to **State 4** (staff notification).
- If it is invalid, increment the failure count. If failures exceed 2, *shake head* and move to **State 3**.

5) State 3:

a) Entry Condition::
Either the user said they have no appointment (from State 1) or they exceeded failures in State 2.

b) Actions::

- Asks the user if they would like to call a human staff member.
- Performs *Speech Recognition* and checks if the answer is positive or negative.

c) Transitions::

- If *yes*, *nod head* and proceed to **State 4**.
- If *no*, *shake head* and return to **State 0**.

6) State 4:

a) Entry Condition::
Indicates the system must notify staff (arrived from State 2 or State 3).

b) Actions::

- Performs *Action: Computer Glance*, then sends an email to the specified staff member.
- Announces that staff has been notified.

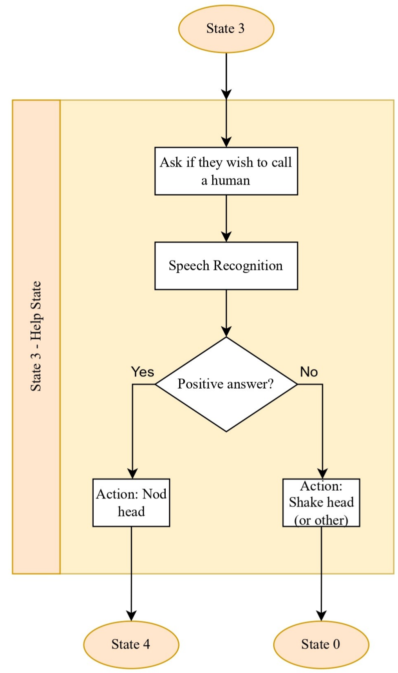

Figure 5. State 3 - Help State

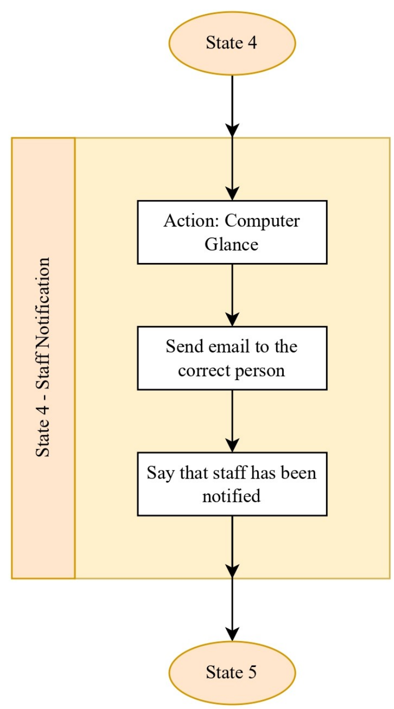

Figure 6. State 4 - Staff Notification

c) Transitions::

- Moves on to **State 5** after notifying staff and confirming the message was sent.

7) State 5:

a) Entry Condition::
Staff have been notified that a visitor is waiting.

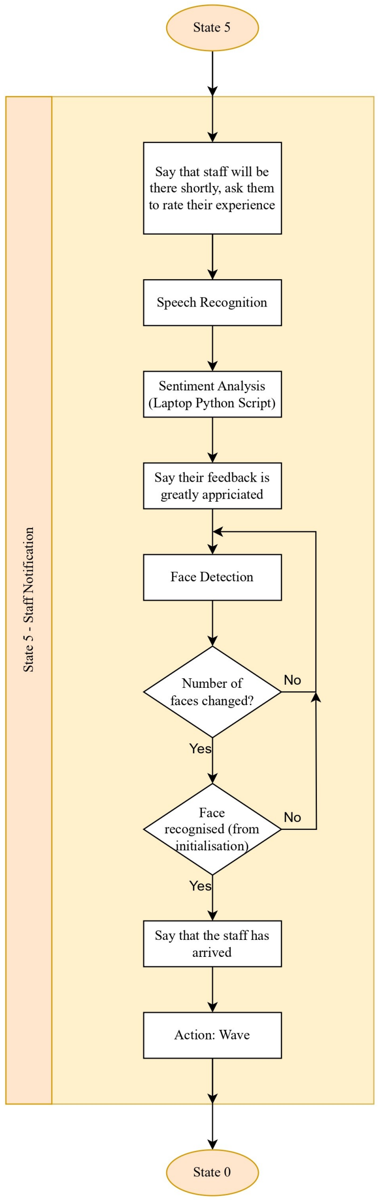

Figure 7. State 5 - Staff Notification

b) Actions::

- Informs the user that staff will arrive soon.
- Asks the user to rate their experience, capturing input via *Speech Recognition*.
- Runs *Sentiment Analysis* to interpret the feedback.
- Thanks the user for their feedback.
- Monitors for any change in faces to detect if staff has arrived.

c) Transitions::

- If the staff's face is now recognised, the system *waves* and returns to **State 0**, ready for the next interaction cycle.

B. Sentences Spoken by the Robot

The robot has several states (**State 0** to **State 5**), each with characteristic announcements or prompts. Below are samples of the main utterances (also visible in the previous subsection):

- **State 1**:
  - `''Hi, my name is Gordon, do you have an appointment?''`
- **State 2**:
  - `''Who do you have an appointment with?''` Prompts for the staff member's name.
  - `''May I ask for your confirmation number?''`
  - `''Sorry, didn't catch that''` If speech recognition is unclear or fails.
  - `''I've got you''` If a matching name/number was found.
  - `''That is not in our database''` If no match is found for either the name or number.
- **State 3**:
  - `''Do you want to call a human?''`
- **State 4**:
  - `''I have sent an email to the member of staff.''`
- **State 5**:
  - Additional prompts about staff arrival and requesting user feedback.

C. Code Overview

The project employs external webcamera vision system for hand gesture detection (wave detection, rude gestures from angry customers), integrating OpenCV and MediaPipe.

1) Main Components:

- **SimpleServer Class:**
  - Handles TCP socket communication, accepting client connections and sending messages.
  - Manages concurrency using threads and locks to ensure safe communication between the robot and external clients.
- **`detectWave` Function:**
  - Utilises OpenCV to capture video frames and MediaPipe to process hand landmarks.
  - Detects waving gestures and rude gestures based on the relative positions of hand landmarks.
  - Sends corresponding signals (e.g., "Wave", "Rude") to the connected client (NAO) via the server.
- **`handle_client` Function:**
  - Receives text input from the client socket.
  - Analyses the sentiment of the received text using NLTK's VADER SentimentIntensityAnalyzer.
  - Sends back a sentiment response (Positive/Negative/Neutral) to the client.

- **`image_processing` Function:**
  - Preprocesses video frames by flipping and converting color spaces for MediaPipe processing.

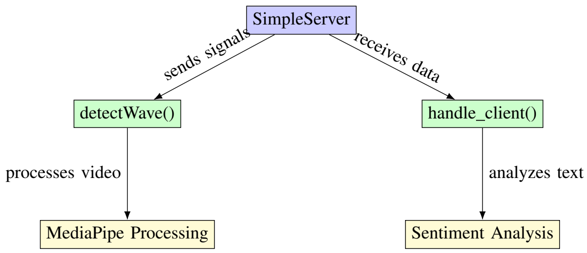

Figure 8. System interaction flow diagram showing main components and their relationships

2) Interaction Diagram:

3) System Workflow:

1) **Initialization:** The main program creates a `SimpleServer` instance, waiting for a client connection.
2) **Wave Detection Loop:** Once connected, the system enters a loop calling `detectWave()`, processing camera frames to detect gestures.
3) **Signal Communication:** Upon detecting a gesture, `detectWave()` spawns a thread to send the appropriate signal (e.g., "Wave", "Rude") via the server.
4) **Client Handling:** Simultaneously, `handle_client()` listens for incoming text data from the client, processes sentiment, and sends back a response.

D. 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.

E. Design Rationale

We placed NAO at eye-level to improve authority and user engagement, as recommended by Rae et al.[4]. This aligns with Sutherland et al.[5], who demonstrated the advantages of a robotic receptionist in clinical settings. Our work builds upon their Wizard-of-Oz study by enabling NAO to generate all behaviours independently.

F. 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.

G. 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.

V. 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.

VI. 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.

VII. 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) **Speech recognition constraints.** NAO struggles with various accents and complex sentences. We could improve this by integrating more sophisticated speech recognition software like Microsoft Azure or Google Speech-to-Text. Using NLP from the NLTK library could also help with understanding complex inputs.
2) **Physical interaction limits.** NAO's movement capabilities can sometimes lead to unclear gestures. We found it's best to stick to simple gestures, and perhaps add a tablet display to help clarify NAO's intentions.
3) **Accessibility concerns.** Healthcare environments serve users with diverse needs. Our current system requires users to see and wave at NAO, which isn't ideal for those with visual impairments. We could implement detection of mobility aids to automatically trigger NAO's check-in process.
4) **User acceptance.** Elderly patients might feel uneasy interacting with robots. Simple visual guides and clear instructions could help make the experience less daunting. We could improve this through more user testing and feedback sessions.
5) **Face recognition reliability.** The face learning and recognition functions proved quite temperamental. Using external cameras and modern algorithms like YOLO could make staff identification more reliable.

C. Module Feedback

The HRI module has been brilliant and engaging. It struck a good balance between theory and hands-on work with NAO. Learning about user-centred design has been particularly valuable for our project work.

The BRL visit was definitely a highlight - seeing real robotics applications really helped put our coursework into perspective.

If we could suggest one improvement, it would be having more time with NAO, perhaps through a booking system or longer lab sessions. This would've helped us get more comfortable with the programming side of things.

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.