SPRING

GRANT AGREEMENT N. 871245

Deliverable D6.3

Robot non-verbal behaviour system in realistic environments

Due Date: 31/01/2022
Main Author: HWU
Contributors: INRIA/ERM/APHP
Dissemination: Public Deliverable

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

DOCUMENT FACTSHEET

Deliverable no.	D6.3: Robot non-verbal behaviour system in realistic environments
Responsible Partner	HWU
Work Package	WP6: WP Learning Robot Behaviour
Task	T6.3: Robot Non-verbal Behaviour System
Version & Date	VFinal, 25/01/2022
Dissemination level	[X] PU (public) [ ] CO (confidential)

CONTRIBUTORS AND HISTORY

Version	Editor	Date	Change Log
V1	HWU	17/01/2022	First Draft
V2	INRIA	21/01/2022	Second Draft
VFinal	HWU	25/01/2022	Final Draft including all partners comments

APPROVALS

Authors/editors	HWU/INRIA
Task Leader	HWU
WP Leader	HWU

Contents

Contents of the Deliverable

1 Introduction

This deliverable presents the initial implementation and software design of the robot non-verbal behaviour system of SPRING, as the main objective of task T6.3. This system is responsible for taking the optimal non-verbal actions in the interaction, based on the dialogue state T5.3 and the overall plan T5.2.

The goal of task T6.3 is to develop the robot non-verbal behaviour system of SPRING that will allow to synthesise the learnt robot behaviour and to choose the appropriate non-verbal actions for the robot to take during interactions. The Robot Non-verbal Behaviour System consists of 2 modules:

- The non-verbal behaviour manager, which interfaces with the high-level planner and conversational system to choose appropriate actions for the robot to take to manage the interactions.
- The robot behaviour generation module, which interfaces with the non-verbal behaviour manager to synthesise the optimal robot behaviour and controls the execution of the robot (non-verbal) actions during interactions.

This document describes the overall architecture for the initial version of the robot non-verbal behaviour system, in Section 2. The non-verbal behaviour manager is described in Section 3, while Section 4 describes the robot behaviour generation module. An example interaction scenario is presented in Section 5, with a report of initial evaluation of the SPRING architecture for a realistic environment.

The software will be released in the [SPRe; SPRf] code repositories. As per European Commission requirements, the repository will be available to the public for a duration of at least four years after the end of the SPRING project. People can request access to the software to the project coordinator at `spring-coord@inria.fr`. The software packages use ROS (Robotics Operating System) to communicate with each other and with the modules developed in the other work packages.

2 Architecture for the Robot Non-verbal Behaviour System

The robot non-verbal behaviour system, which consists of the Non-verbal Behaviour Manager and Robot Non-verbal Behaviour Generation modules, is responsible for choosing the appropriate non-verbal actions to take and synthesise robot behaviour enabling multi-modal multi-person interaction and communication.

Non-verbal behaviors in the SPRING project are comprised of navigation and gesture (mainly arm, head, and eye movements) behaviors, that will allow the robot, for instance, to: navigate towards a person or a group of people to start a conversation; explore the environment while avoiding obstacles; move towards one or several persons in order to improve the quality of the sensory data (images and acoustic signals); attract the attention of the selected persons in order to facilitate face-to-face communication; and other action policies for multi-party conversation management.

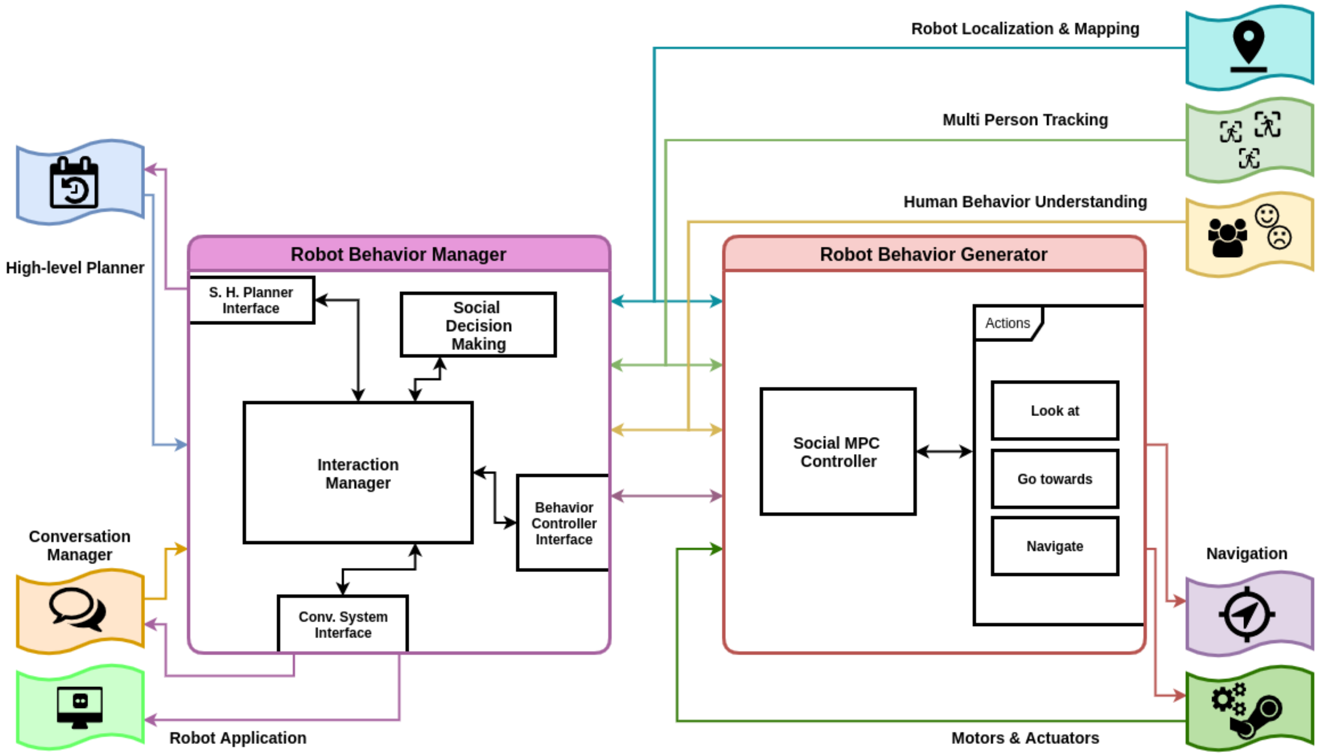

Figure 1: The Robot Non-verbal Behaviour System, which consists of the Non-verbal Behaviour Manager and Robot Non-verbal Behaviour Generation modules allowing to synthesise robot behaviour and to choose the appropriate non-verbal actions. The non-verbal behaviour manager, interfaces with the high-level planner and conversational system, and is responsible for deciding appropriate actions to take to manage the interactions. The robot behaviour generation module interfaces with the non-verbal behaviour manager and controls the execution of the robot (non-verbal) actions during interactions. The Robot Non-verbal Behaviour System modules take as input the high level information provided by the other modules in SPRING, such as: robot localization and mapping, multi-person tracking, human behaviour understanding, conversation manager, high-level planner, and robot motors' and actuators' joint states. The outputs are given as robot actions, in the form of motor and actuator commands for gestures and navigation, as well as updates for the high-level plan and conversational dialogue state, and the robot application on the ARI tablet screen.

Many of these tasks are complex and based on high-dimensional variables such as the camera or auditory input from the robot. The Non-verbal Behaviour Manager and Robot Non-verbal Behaviour Generation modules take as input the high level information provided by other work packages: robot localization and map of the environment (WP2), human behaviour understanding (WP4), localization and identification of people with respect to the robot (WP3), dialogue state and the overall plan state (WP5), robot joint states (WP7). The outputs are given as robot actions, in the form of motor and actuator commands for gestures and navigation, as well as updates for the high-level plan and conversation dialogue state.
The overall architecture of the robot non-verbal behaviour system is illustrated in Fig 1. A description of the the Non-verbal Behaviour Manager and Robot Non-verbal Behaviour Generation modules is given in the next sections. Section 3 presents the non-verbal behaviour manager, whereas in Section 4 we describe the robot behaviour generation module.

3 Robot Non-verbal Behaviour Manager

The non-verbal behaviour manager is responsible for deciding appropriate high-level social actions to take and to manage the interactions. The social interactions in SPRING required the behavior system components to interface with social perception signals, a task planner, and a multi-user conversational manager, to enable situated interactions with multiple users at the same time.
The robot non-verbal behaviour manager consists of the Interaction Manager and the Social Decision Making components, and handles the interface between the high-level planner, the conversational system and the robot behaviours. As shown in Fig 1. The robot behaviour manager module interfaces with the non-verbal behaviour generation through the Robot Non-verbal Actions servers. The behaviour manager handles high-level interaction decisions and the behaviour generation module controls low-level action execution.

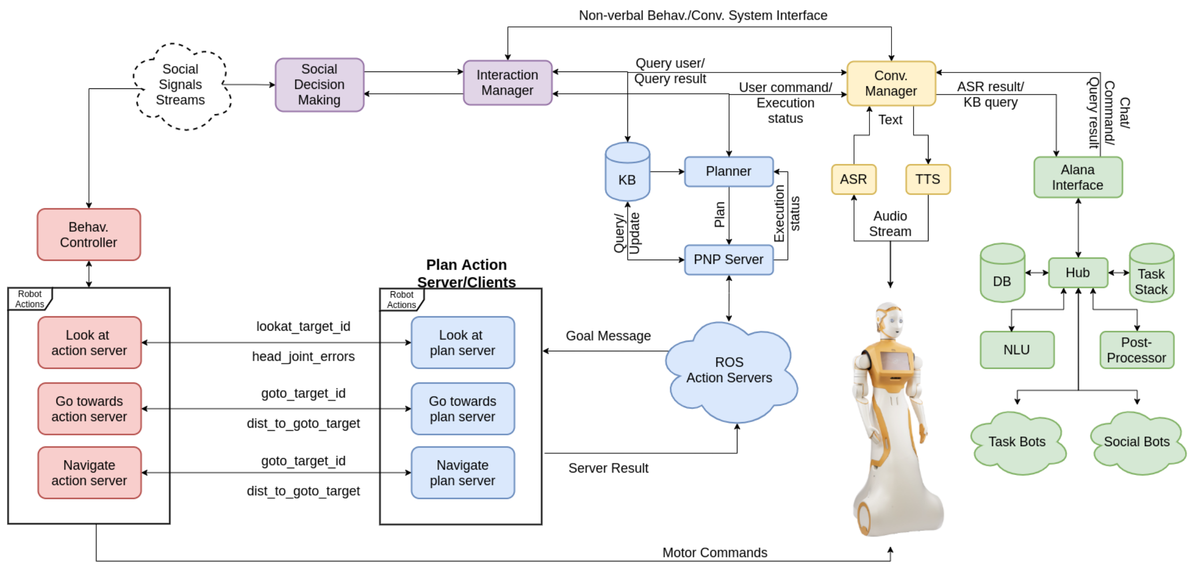

Figure 2: System architecture for the Robot Non-verbal Behaviour System with the Initial High-level Task Planner and Conversational System (extending figure 1 from SPRING Deliverable 5.1 [SPRb]). The non-verbal behaviour manager (in purple) the high-level planning framework (in blue), the conversation manager interface (in yellow) and the conversational system on the right (in green) with the robot non-verbal behaviour manager components (in red) on the left.

3.1 Interaction Manager

The Interaction Manager handles the interface between the Robot Behaviour Generation modules (WP6) and the High-level Robot Task Planner and Conversational System (WP5). As shown in Fig 2.

3.1.1 High-level Planner Interface

The Interaction Manager is implemented as an abstract controller for the ROS Petri-Net Planner presented in SPRING Deliverable 5.1 [SPRb]. The controller can start tasks and keeps track of them. The main functionality of the controller is to manage the currently available and running Petri-Nets and to provide functionality to send and receive information from/to a specific net. For taking care of the running Petri-Nets plans, the controller interface provides information on the available nets. The other major functionality of the controller interface is to exchange data between the different plans and the social state representation provided by the Social Decision Making component.

3.1.2 Conversation Manager Interface

Both the Interaction Manager and the Conversation Manager are implemented as abstract controllers for the ROS Petri-Net (RPN) Planner presented in SPRING Deliverable 5.1 [SPRb]. Each controller is in charge of its own type of RPN planner servers. In order to communicate with the controllers, we a use a knowledge base implementation. The Conversation Manager Interface handles communication between the conversational system and the non-verbal behavior system in order to maintain, synchronize, and up date knowledge about the dialogue state and the interaction state for each controller.

3.1.3 Behaviour Generator Controller Interface

The non-verbal behaviours of the robot in SPRING are implemented as a collection of ROS Actions Servers, the Robot Behavior Manager will handle/synchronise the different ROS Actions Servers from the Robot Behavior Generator. These are integrated through ROS [Sta18] action server/client interface, as shown in Fig 2, for each robot non-verbal action/plan.

3.2 Social Decision Making

In the SPRING project we need to be able to understand various individual and group situations and take appropriate decisions, e.g. identify persons that need assistance (such as patients that have been waiting for a long time and who might be anxious), engage in face-to-face multi-modal dialogue with a patient, a family member, a staff member, or with a party of them, or accompany a patient across the hospital.

Neural architectures for the social decision making interaction behavior were described in SPRING Deliverable 6.1 [SPRc]. The Social Decision Making component is tasked with turning the continuous stream of messages produced by the low-level input and output components into a discrete representations to describe multi-party interactions, devise social interaction plans, and support the high level planner and the conversational manager for maximizing the robot's execution strategies for social interaction and communication.

An initial set of "social decisions" that are require for the SPRING robot non-verbal behaviour system to make for interactions in a realistic environment, as part of the the initial software design and implementation, have been summarize in Table 1. More complex actions and decisions, such as 'joining and talking with a group' will be added in a later stage.

Table 1: Robot Non-verbal Behaviour System initial set of "social decisions"

Decision	**Description**	**Action**
Detect arrivals	People arrive in front of the robot, ARI detects their arrival and goes towards them to start an interaction	Start interaction
Detect departure	Detects the departure of people and returns to original position	End interaction
Detect attempts to interact with it	ARI determines a person in the scene wants or requires attention	Start interaction
Detect (groups) people interacting	ARI determines a group of people are busy engaging with themselves and withdraws for then continue the discussion without the robot	End interaction /Non interfere
Detect/Recognize staff members	ARI recognizes a member staff is interacting with the person(s) and withdraws to not interfere with staff functions	End interaction /Non interfere
Decide who/what to look at	Decide who to look at, switch focus of attention during multy-party interactions	Realign position /orientation
Decide to start/go navigation	Decide when to go, start an approach or guidance action, adapt to persons in scene	Approach/Guidance
Decide what to show on tablet	At different times during interaction, ARI should display message on its screen, a welcome screen, the weather, map, etc.	Display to screen

4 Robot Non-verbal Behaviour Generator

The robot non-verbal behaviour generation is in charge of the synthesis and execution of the robot (non-verbal) actions during interactions. Controlling the positioning, navigation and gestures execution of the SPRING robot. The robot non-verbal behaviour generation module consist of the Social Model Predictive Control and a collection of ROS Actions Servers, one for each of the Robot Non-verbal Actions types required. As shown in Fig 1.

The robot behaviour generation module interfaces with the non-verbal behaviour manager through the Robot Non-verbal Actions servers. The behaviour generation module controls low-level action execution and the behaviour manager handles high-level interaction decisions.

4.1 Social Model Predictive Controller

The low-level robot control for the robot non-verbal behaviour generation is handle by a Social Model Predictive Control (MPC) controller. The general architecture for the MPC controller of the robot is described in SPRING Deliverable 6.1 [SPRc].

4.2 Robot Non-verbal Actions

The non-verbal behaviours of the robot in SPRING are implemented as a collection of ROS Actions Servers in the Robot Behavior Generation that are triggered/synchronised by the the Robot Behavior Manager, as shown in Fig 2.

The behaviours (non-verbal) of the SPRING robot in the robot non-verbal behaviour system for a realistic environment for the initial software design and implementation include aligning ARI to face a direction to look at people, or turn in the direction of a person or object, displacing ARI to a location to move around a room, approaching a person to join a conversation, guiding patients through the clinic, going to its base charge station, and arm/head motions for gestures such as pointing, waving, etc. These have been synthesized into 3 robot non-verbal actions: 'Look at', 'Go Towards', 'Navigate to'. Additionally a set of pre-defined gestures is available from the robot software architecture [SPRd].

4.2.1 'Look at' Action

The 'Look at' action aligns ARI to face a direction, specified in the command, to orient ARI's body to look at a target, such as a person during an interaction. The 'Look at' action requires a *target_id* to look at (person ID) and a timeout period in seconds when the actions should have ended. It returns a *head_joint_errors* message, with the *final_time* it took to execute the action.

4.2.2 'Go Towards' Action

The 'Go Towards' action moves ARI to a location, specified in the command, to approach a person or group to start an interaction, or go to a location in the room. The 'Go Towards' action requires a *target_id* from the person or group to approach and a timeout period in seconds when the actions should have ended. It returns a *dist_to_goto_target* message, with the *final_time* it took to execute the action.

4.2.3 'Navigate to' Action

The 'Navigate to' is a guidance task to escort a patient through the clinic. The 'Navigate to' action requires a *goto_target_id* (localization ID) and *human_target_id* (person ID), as well as a *goto_target_pose*, and a timeout period in seconds when the actions should have ended. It returns a *dist_to_goto_target* message and *dist_to_human_target*, with the *final_time* it took to execute the action.

5 SPRING Robot Behavior and Planning Interaction Scenario

A demo scenario use case, for the SPRING robot behavior and planning interaction scenario in a realistic environment, has been developed for the preliminary software architecture validation in controlled environments (laboratory setting).

5.1 Scenario Description

Location: SPRING ARI Robot is located at the waiting room area of the *'day hospital'*.

Robot Role: The SPRING ARI Robot will perform as a *'reception assistant'*. Functions will include **"welcoming"** patients/visitors to the waiting room, **"answering"** patients' queries, providing **"orientation"** and **"guidance"** to patients/visitor to the hospital, and providing **"entertainment"** to patients easing and improving their waiting room experience and liberating hospital staff from interruptions, repetitive request, distracting task, etc., that takes away from their main goals and could be done by the robot.

Robot Goals: (main) Monitor the waiting room of the *'day hospital'*:

1. Detect/interact with new patient arrivals and welcome them to the *'day hospital'*.
2. Detect/interact with reentries to the waiting room and assist them if needed.
3. Detect patients wishing to interact or trying to ask questions.
4. Detect patients who may be distressed/uncomfortable/bored in the waiting room and attempt to engage them in conversation.
5. Answering common general questions about the *'day hospital'* procedures and the waiting room.
6. Provide orientation and/or (verbal) guidance, to patients who require so, about locations/services in the *'day hospital'*.
7. Communicate/alert members of staff when patients need assistance.
8. Not interfere with members of staff when performing their duties.

Robot High-level Plans:

- Wait (monitor)
- Start Interaction (approach)
- Interact (conversation)
- End Interaction (return to wait)

wait (monitor)

waiting_room_robot_idle_monitor

initial knowledge:
⟨ start_location ⟩ robot_base_waiting_position

plan:
- while:
condition: nothing_happens
actions:
- surveil_mode
(look around the room, in place or roaming)
- monitor_and_decision_making
Two people arrive in front of the robot (New arrivals)
PersonA approaches ARI and utters, "Hey ARI."
Patient comes in front of the robot
(something_happens ⟨1,2,3,4⟩)

- if condition: detected ⟨1,2,3,4⟩
actions:
- start_interaction

- end_go_back_to_base

start interaction (approach)

robot_approaching_user

initial knowledge:

⟨ start_location ⟩ robot_location
⟨ target_user ⟩ [userID]
⟨ target_group ⟩ user's group

plan:

- **while:**
  condition: robot_wants_to_interact_with_target
  (start_interaction & !robot_is_interacting & !approaching)
  actions:
  - **start_approach ⟨target⟩**
    calls for robot 'Go Towards' Action

- **if condition: success**
  actions:
  - **interact_with_user**

- **if condition: failure**
  actions:
  - **end_go_back_to_base**

interaction (conversation)

robot_interacting_with_users

initial knowledge:

⟨ target_user ⟩ [userID]
⟨ target_group ⟩ user's group
⟨ interact_goal ⟩ (1),(2),(3),(4)

plan:

- **start_conversation_with_target**
- **while:**
  - **condition: robot_is_interacting**
    (there's a user engage with the robot)
  - **actions:**
    - **if !goal_of_interaction**
      actions:
      - **conversation_action_towards_goal**
        e.g. If goal to welcome a patient
        and this requires them taking a ticket,
        robot repeats the request if this has not been met.
    - **if conditions_that_should_alter_interaction**
      actions:
      - **set_conditions_for_behaviour_as_needed**
        The conversation manager will activate plans
        as needed from the conversation state,
        which will execute if conditions are right
        e.g. gesture, pointing, look at, display, etc.
- **end_interaction**

end interaction (return)

```
robot_end_interaction_return_to_base

initial knowledge:
    ⟨ start_location ⟩ robot_location
    ⟨ base ⟩ robot_base_location

plan:
    - while:
        condition: robot_is_interacting
         (there's a user engage with the robot)
        actions:
            - stop_interaction_with_users

    - go_back_to_base

    - go_to_waiting
```

5.2 Example Scenario

In the example interaction scenario two people arrive in front of the ARI robot located at the waiting room area of the *'day hospital'* and start an interaction. (HLP = High-level planner — RBM = Robot Behavior Manager — RBG = Robot Behavior Generation — RA = Robot Application — DM = Dialogue Manager — PM = Perception Modules)

- ∗ Two people arrive in front of the robot (**new arrivals**)
- ∗ **ARI** detects their arrival ⇒ PM/RBM
- ∗ **ARI** looks at them. ⇒ RGB
- ∗ **ARI** decides to go to interact with these people ⇒ HLP/RBM
- ∗ **ARI** goes towards them ⇒ RBG/RBM
- ∗ **ARI** displays a welcome screen ⇒ RA/RBM
- ∗ **ARI** says a welcome message: "Hello, I am ARI. I can give you some information about how the service works." ⇒ DLM
- ∗ **PersonA** replies: "Thank you ARI. We have several questions."
- ∗ **ARI** must look at **PersonA** (**ARI** decides to look at **PersonA** (HLP/RBM), **ARI** looks at **PersonA** (RBG/RBM)
- ∗ **ARI**: "I am here to help you. I am listening."

- ∗ **PersonB**: "This is the first time we come to Broca hospital. What is the course of the day?"
- ∗ **ARI** turns to **PersonB**, look at her and nods (RBG/RBM): "You will have several appointments with different professionals, at least three. First, a nurse will ask you questions about your health and will make an initial assessment of your current condition. Secondly, you will see a neuropsychologist with whom you can discuss your difficulties and who will suggest tests to assess your abilities as accurately as possible. Thirdly, you will meet a geriatrician who will synthesise all these data and propose different types of follow-up depending on your results."
- ∗ **PersonA**: "Thank you ARI."
- ∗ **PersonB**: "Thank you ARI. That was really clear."

* **ARI**: "Please, I am here to help you. Do not hesitate to ask me if you have any other questions."

* **PersonB**: "Where are the consultations held?"
* **ARI** look at **PersonB** and nods (RBG/RBM): "The professionals come to pick you up in this room and welcome you in their office."
* **PersonA**: "Ok, thanks."
* **PersonA**: "What will be our first consultation?"
* **ARI**: "That's a good question, it depends on the availability of the nurses. You can meet either the nurse or the neuropsychologist. These two appointments always precede the appointment with the doctor."
* **PersonB**: "What is the work of the neuropsychologist?"
* **ARI**: "The neuropsychologist seeks above all to establish your difficulties of memorization, expression and reasoning. This helps the doctor to propose the best possible follow-up. To do this, he proposes a series of targeted exercises that informs him about your aptitudes."
* **PersonB**: "Thank you ARI."
* **ARI**: "You're welcome."

* **PersonB** : "I have one more question. Do we have a breakfast and a lunch?"
* **ARI**: "Of course, a lunch will be proposed to you at mid-day if you have consultations in the afternoon. A snack can be offered after your appointment with the nurse."
* **PersonA**: "ARI, do you know who takes care of the meals?"
* **ARI**: "Meals are included in the price of your consultation day. You don't have to pay anything extra."
* **PersonB**: "Very good. Thank you ARI."
* **ARI**: "You are welcome. I am here to answer all your questions."

* **PersonA**: "Thank you ARI. That's very kind of you. Can you point us to the bathroom?"
* **ARI** has to look at **PersonA** ⇒ RBG/RBM
* **ARI**: "Very easy! You have to go to the corridor at the end of the room. It's the second door on the right."
* **PersonB**: "Thanks ARI. We're going in."
* **ARI** turns to **PersonB** ⇒ RBG/RBM
* **ARI**: "Please. Feel free to come ask any more questions. I have plenty of information to give you."
* **ARI** detects the departure of the two people ⇒ PM/RBM and returns to home position ⇒ HLP/RBM/RBG

5.3 Validation of the robot non-verbal behaviour system in realistic environments

To validate the robot non-verbal behaviour system, a series of experiments with participants were carried out, under laboratory conditions, at the research facilities of HWU and INRIA.
At HWU facilities, a formative user evaluation was carried out. Seven participants were recruited and asked to interact with the SPRING robot on a set of predetermined tasks, and to give their opinions of the interaction with the system. Particpants were supplied with a fictitious persona, in a hospital 'waiting room', and asked to carry out three tasks (asking for directions (D), for information on catering arrangements (C) and for schedule information (S)) to provide them with experience of the system's non-verbal behavior manager, task planner and the conversational system. After interacting with the robot, participants completed a user attitude questionnaire about the experience. User feedback was positive towards many aspects of the robot and the interaction, users did not feel 'anxious', found robot 'enjoyable', 'sociable' and 'would be happy to talk to it again'.
At INRIA facilities, three experimental trials with two participants per trial were conducted. In each trial, ARI had to navigate twice to a single participant and twice to a group of two participants, to evaluate the interface between non-verbal behavior generator and non-verbal behavior manager in an interaction, and the non-verbal behavior generator, 'Go Towards' action (Section 4.2.2). Under each condition (navigate to a single participant or a group), the direct path was either blocked by an obstacle or free. The obstacle was a low table around which ARI had to navigate to reach the participant or group. The results showed that the non-verbal behavior generator was successfully able to navigate ARI to its targets. Furthermore, a questionnaire about the participants perception of ARI's behavior during the experiments showed that participants were not afraid to interact with ARI. Importantly, the participants rated that the distance and speed of the robot was comfortable.
A full report of the validation experiments and results was reported on SPRING Deliverable 1.4 [SPRa]

6 Implementation

The software modules describe in this deliverable will be made available on the SPRING project Gitlab repositories for Work Package 5 [SPRe] and Work Package 6 [SPRf]. The software packages use ROS (Robotics Operating System) to communicate with each other and with the modules developed in the other work packages.
These will be available to the public for the duration specified in the SPRING project proposal. As per European Commission requirements, the repository will be available to the public for a duration of at least four years after the end of the SPRING project. People can request access to the software from the project coordinator at `spring-coord@inria.fr`.

References

[SPRa] SPRING Project. *D1.4: User feedback from the preliminary validation (realistic environments)*. URL: https://spring-h2020.eu/results.

[SPRb] SPRING Project. *D5.1: Initial High-level Task Planner and Conversational System Prototype for Realistic Environments*. URL: https://spring-h2020.eu/wp-content/uploads/2021/06/SPRING_D5.1_Initial_High-level_Task_Planner_and_Conversational_System_Prototype_for_Realistic_Environments_vFinal_31.05.2021.pdf.

[SPRc] SPRING Project. *D6.1: Neural network architecture specification and design*. URL: https://spring-h2020.eu/wp-content/uploads/2021/03/SPRING_D6.1_Neural-network-architecture-specification-and-design_VFinal_24.02.2021.pdf.

[SPRd] SPRING Project. *D7.3: Initial software architecture for SPRING-REEM*. URL: https://spring-h2020.eu/results.

[SPRe] SPRING Project. *WP5: Spoken Conversations*. URL: https://gitlab.inria.fr/spring/wp5_spoken_conversations.

[SPRf] SPRING Project. *WP6: Robot Behavior*. URL: https://gitlab.inria.fr/spring/wp6_robot_behavior.

[Sta18] Stanford Artificial Intelligence Laboratory et al. *Robotic Operating System*. Version ROS Melodic Morenia. May 23, 2018. URL: https://www.ros.org.