SPRING

GRANT AGREEMENT N. 871245

Deliverable D5.1

Initial High-level Task Planner and Conversational System Prototype for Realistic Environments

Due Date: 31/05/2021
Main Author: HWU
Contributors: UNITN, CVUT, BIU, 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.	D5.1: Deliverable Name
Responsible Partner	HWU
Work Package	WP5: Multi-User Spoken Conversations with Robots
Task	T5.1: Initial Conversational Prototype
Version & Date	VFinal, 17/05/2021
Dissemination level	[X] PU (public) [ ] CO (confidential)

CONTRIBUTORS AND HISTORY

Version	**Editor**	**Date**	**Change Log**
V1	HWU	01/05/2021	First Draft
VFinal	HWU, INRIA	17/05/2021	Final Draft including all partners comments

APPROVALS

Authors/editors	HWU + UNITN, CVUT, BIU, APHP, INRIA
Task Leader	HWU
WP Leader	HWU

Contents

Contents of the Deliverable

1 Introduction

The overall objectives of WP5 (Multi-User Spoken Conversations with Robots) are to develop a) techniques for multi-user conversation involving a robot and multiple humans, and b) the overall robot task planning.

This deliverable presents the implementation of an initial dialogue system, as the main objective of task T5.1. This initial conversational prototype will be used to facilitate data collection. The modules presented in this document have been implemented, and have been interfaced using a web-based chat interface[1] to allow testing and validation. This was done to mitigate the impact of the COVID-19 pandemic on the project's progress, as constraints due to the pandemic still impede conducting experiments in our research laboratories.

This document describes the overall architecture for the initial version of the conversational system and high-level task planner, in Section 2. The high-level planner is described in Section 3, while Section 4 describes the conversational system. A set of example conversations is presented in Section 5. Section 6 ends with suggestions for future developments and research work for the remaining tasks in WP5.

The software is being updated in SPRING-WP5-Repository. 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 workpackages.

2 Architecture for the Initial Conversational Prototype

The first goal for WP5 is to develop and implement an initial version of the conversational system and the high-level task planner. The conversational system provides initial functionalities of Natural Language Understanding (NLU), Generation (NLG), and the Dialogue Manager (DM). The high-level task planner allows for concurrent execution of dialogue and task-based actions based on the current dialogue, and interaction status. The high-level task planner will decide which is/are the forthcoming task/s to be performed by the robot. These components will later on be enhanced in tasks T5.2 and T5.3, and connected to the non-verbal behaviour manager of WP6. The initial system has been implemented for the first integration cycle T7.3 and to collect the data of T1.2 for training data-driven modules.

- The **conversational system** is an extension of the social bot Alana, which was twice finalist in the Amazon Alexa challenge [Papaioannou et al., 2017a,b, Curry et al., 2018] and has previously proven successful as a foundation for other conversational AI projects [Foster et al., 2019].
- The **high-level task planner** is implemented by a Petri Net planner [Dondrup et al., 2019], and interfaces the dialogue system and the sensors and physical actions of the robot. Thus, based on the detected user's intent, it executes a plan combining dialogue, physical, and perception actions.

The overall architecture of the prototype system, illustrated in Fig 1. Section 3 presents the initial prototype of the high-level task planner, whereas in Section 4 we describe the initial conversational system.

3 High-Level Task Planner

In SPRING, the planning and execution framework comprises mainly a planning component (3.2), which produces sequences of actions that need to be carried out in order to fulfil a given goal, and an execution component, based on automatically generated finite state machines (PNP) (3.3), to execute these actions. The high-level task planner

[1] Available at `http://152.67.132.99:3000/?#/visual_dialog`

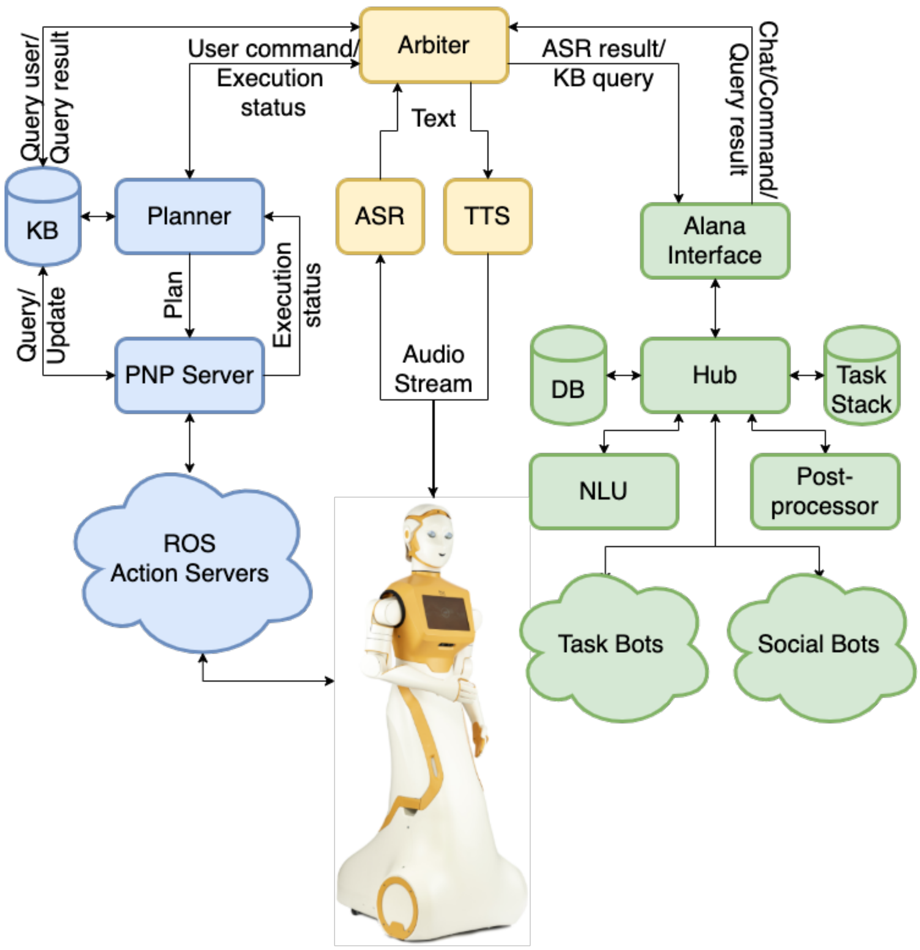

Figure 1: System architecture for the Initial High-level Task Planner and Conversational System Prototype: showing the planning and execution framework on the left (in blue) and the conversational system on the right (in green), and components (in yellow) combining the two systems with the robot in the middle.

will be based on the ideas of Lemon et al. [2002] and the system presented in Papaioannou et al. [2018] to allow for a flexible, pausable, and multi-threaded task execution and dialogue. As can be seen in Fig. 1, the planning and execution framework is comprised of several distinct components, i.e. the planner, the Petri-Net Plan server, the knowledge-base (KB), the set of ROS action servers, and the arbiter node which combines the planning and execution framework with the dialogue management concepts of Section 4.

3.1 Arbiter

The Arbiter manages the communication between the conversational system, the robot, and the planner. There are three distinct cases:

(i) input from the speech recogniser (ASR) is always forwarded to the conversational system (Alana),

(ii) output from the conversational system (Alana) might go to the speech synthesizer (TTS), the planner, or both,

(iii) output from the planner is forwarded to the conversational system (Alana).

In addition to managing the communication, the Arbiter also tracks all the currently running plans/tasks and sends Alana's output to the task that required this information.

3.2 Recipe-based Planner

Whenever a new task from the robot is required, the Arbiter described above will instruct the Planner, in Fig 1, to create a new plan to fulfil the required task.
The system described here uses the principle of *recipes and resources* as developed in Lemon et al. [2002] to eliminate the problems from (re-)planning delays and concurrent interactive planning and execution. Similar to PDDL, a domain file is defined that lists all the possible actions and defines their types and parameters. These recipes use simple YAML files which describe a sequence of actions, the tasks involved in achieving a given goal, and the resources required to execute these tasks.
The plans support all the usual logical operations, i.e. `and`, `or`, and `not` and also generic python operations. In addition to sequencing single actions, the petri-net is able to execute actions concurrently by specifying a list of `concurrent_actions` which can themselves also contain other concurrent actions. Also loops can be easily constructed using the `while` keyword.

3.3 Petri-Net Plan Server

The sequence of actions described in a plan are transformed into a Petri-Net Plan (PNP) [Dondrup et al., 2019], which allows for the addition of recovery behaviours automatically generated from the preconditions and effects of the actions and the concurrent execution of multiple actions simultaneously. This module allows execution of the actions quickly and robustly using a finite state machine built automatically using PNP. This combines the flexibility of the planner with the speed and robustness of an FSM. This framework provides an easy to use system with a native integration of ROS action servers.
The recipes will be transformed into PNPs if and only if the resources are currently available (as stored in the KB) and can then be concurrently executed together with other plans that might already be running. Using the ability of the PNPs to include precondition and effect checks together with recovery behaviours, it is trivial to skip redundant actions whose effects have already been achieved or define repair actions in case an action was unsuccessful and, therefore, eliminate the need for constant re-planning.
Before executing an action and after its execution, certain behaviours are automatically inserted to check conditions and recover from execution errors. Before the start of an action, the local KB is queried to check if the variables required to fill the goal message are present. If they are not, then it reports a failure. After the execution of an action sever outcomes are checked. If the execution was successful the PNP continues as expected. If the action failed, it reports a failure or if the action is interrupted, followup recoveries can be defined or the PNP can continue regardless.
In addition to being able to concurrently execute actions, these plans also allow to create loops and thanks to the mentioned recovery behaviours if-then-else constructs. The system also allows to concurrently execute several Petri-Net Machines (PNM).

3.4 Knowledge-Base

The knowledge-base (KB), as part of the PNP Server, is split into two parts, i.e. the local and remote KB.
One of the problems of state machines is the passing of knowledge between states. For our execution framework, we use a local KB that fills all data fields in the goal of the action to be sent to the server automatically and is updated by the resulting data produced by an action server after its execution finished. Moreover, by creating a disjoint instance of the local KB for each PNM that is executed, the concurrent execution of several plans each having their own KB makes it thread safe.
The framework can also be interfaced with a number of global KBs like a data centre to query information or a dialogue system. The remote KB will hold globally relevant information about the interaction/the world and provides a means to query information communicating with the human via the Arbiter.

3.5 Action Servers

The defined domains and plans are translated into action servers themselves which, when started, execute the plan as a PNM. Once a plan has been loaded, it becomes a ROS action server itself for ease of use. Actions are defined with a name, a list of parameters, preconditions, and effects. Preconditions and effects can use logical operations. The actions are executed in order and can be given arguments for the parameters they expect. If any of the parameters is omitted, it is filled from the KB automatically.

The new Petri-Net server supports 3 different types of actions to be executed: 1) standard ROS action servers, 2) ROS Petri-Net (RPN) action servers which allow the action server to query and update the knowledge base at run-time, and 3) KB actions that just query and update the KB. Each action has 3 different outcomes, i.e. succeeded, preempted, and failed, which can all be associated with recovery behaviours like failing the plan, restarting the action, executing other actions, etc. Each action also has preconditions and effects which are checked before and after the action has been executed.

4 Conversational System

We here present the architecture of the initial conversational system for this project (see Fig. 2).
The original Alana system is an ensemble of two types of chat bots, 1) functional bots that drive the conversation in case it has stalled, deal with profanities, handle clarifications, or express the views, likes, and dislikes of a virtual Persona; and 2) content-delivery bots that might potentially produce a reply to the user's utterance from different sources, e.g., Wikipedia, Reddit, and different News feeds as well as a database of interesting facts. The decision on which bot's reply is selected to be verbalised is then handled by the Dialogue Manager (DM).

- **Natural Language Understanding (NLU)** – in the Alana system, users' utterances are parsed using a complex NLU pipeline, described in detail by Curry et al. [2018], which consists of steps such as Named Entity Recognition, Noun Phrase extraction, co-reference and ellipsis resolution, and regex-based intent recognition. In the SPRING project, we have also employed the RASA framework[2] to handle enquiries specific to the hospital domain, such as check-in procedures and directions to locations within the building. In addition, the Persona and Quiz bot rely on the scripting language AIML[3] or similar patterns to parse the user queries.
- **Natural Language Generation (NLG)** – the NLG strategy is designed for each bot. In the initial system, it mainly employs complex and carefully designed templates for different bot-specific purposes, or else, content retrieved from "safe" online sources such as Wikipedia and News sites.
- **Dialogue Management (DM)** – at every dialogue turn each bot attempts to generate a response. The decision as to which response is selected is then handled by the Dialogue Manager. The current selection strategy is based on a bot priority list, although it can also be learned from data [Shalyminov et al., 2018]. If multiple bots produce a response, the response from the bot with the highest priority is selected. In the initial SPRING system, the Visual Engagement and Task Bots have highest priority, followed by the domain-specific chat bots: Reception Bot, Directions Bot, Corona Bot and Quiz Bot, in that order. The remaining bots follow the order of priority employed by Curry et al. [2018].

4.1 Domain-Specific Conversations

In order to meet the medical care requirements, the initial SPRING system extends the original Alana pipeline by keeping the basic functional bots (see blue blocks in Fig. 2) and then introducing several domain-specific bots (as described below).

4.1.1 Reception Bot

The Reception Bot provides practical support to patients and their companions who are visiting the Memory Clinic within the Broca hospital clinic. It is responsible for greeting visitors, supporting them during check-in and answering frequently asked questions relating to the clinic's facilities (e.g. "is lunch provided?") and the format of the patient's visit. For example: User: "Can I check in?", Robot: "Welcome, I can help you with that. May I have the name please?"

4.1.2 Directions Bot

The Directions Bot handles requests for help in finding key locations and/or facilities within the clinic itself, and the wider hospital, e.g., bathrooms, the way out, the coffee machine, etc. The directions are currently written in ab objective style, relating to the position of locations relative to the room where the robot will be deployed, for

[2] https://rasa.com
[3] http://aiml.foundation

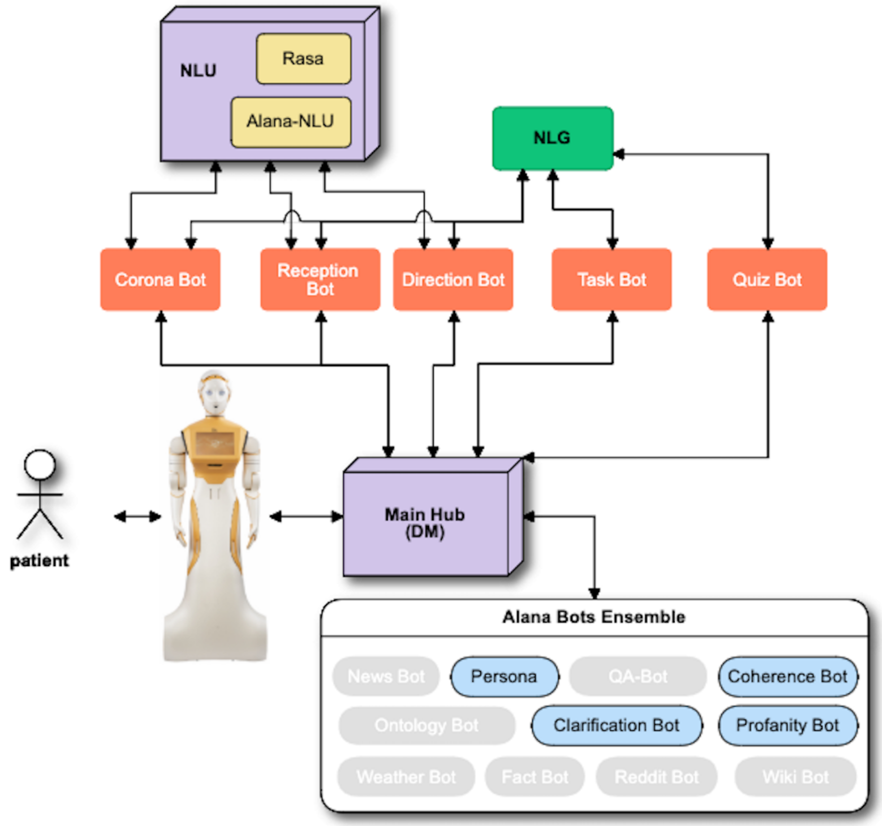

Figure 2: Conversational System Architecture. The social chat bots (grey blocks) have been disabled for the initial SPRING system. Instead, we present several domain-specific conversational bots (orange blocks).

example: "When you leave the dining room, turn right and go through the doors with two round windows. Then go right towards the waiting room. The reception desk will be in front of you."

4.1.3 Task Bot

The task bot is able to trigger tasks and handle communication between the task and the user. As described in section 3.1, Alana communicates with the robot using the Arbiter node. As such, the task bot exchange information with the planning and execution framework via the inclusion of specifically formatted commands in the response which, besides text to be synthesised via the TTS, hold all the information required for the next task dialogue act (as opposed to the social dialogue sub-bots that require and respond using text messages only).

4.1.4 Corona Bot

The Corona Bot is a natural language bot designed to inform and support members of the public during the Covid-19 pandemic in a number of ways (see dialogue example in Table 2). Firstly, it provides direct access to key facts relating to the virus itself (e.g., methods of prevention, symptoms, risks, vaccination). Secondly, it tackles misinformation by rebutting commonly circulated myths regarding the virus (e.g., its origins, methods of transmission and potential "cures"). Finally, it provides mental health support, offering tips on how to handle, e.g., boredom, loneliness and low mood during the pandemic. In each case the information supplied is synthesised from highly reliable sources, such as the World Health Organisation (WHO), the U.K.'s National Health Service and mental health charities.

4.1.5 Corona Quiz Game

The Quiz game is a hand-crafted chat bot that employs a XML-styled conversational flow. The bot is deployed to keep patients entertained while they wait, by answering a list of topic-specific questions (e.g., Covid-19). Those

particular questions have been chosen to draw attention towards common tips provided by the WHO, which aim at helping to manage an individual's physical health.
The content of the quiz can easily be modified in the future for example for music, general knowledge, etc.

4.1.6 Visual Dialogue

As well as using the list of domain-specific bots above, receptionist robots will need to be able have visual skills, such as to greet visitors, offer them a place to sit, answer basic questions they may have, show them where the bathrooms are, help them locate a missing object, and point them where to get food and drinks. One important capability that is required to fulfil most if not all of these tasks is being able to talk about the surrounding environment (so-called 'visual dialogue'), e.g. recognising objects, identifying their locations, their attributes, and their relative positions, etc.
To achieve such functionality, the dialogue system processes the user's utterance, inferring the user's intent and any entities the user may have mentioned. This information is then forwarded to the visual action server in the form of a goal message. Upon reception of the goal message, the server accesses a graph of the scene in order to answer questions such as: "have you seen my purse?" or "where can I get something to drink?". In the initial system, these graphs have been manually created based on the scene segmentation results. However, we are working on generating these graphs automatically through the combination of various automatic scene graph generation pipelines and pruning and refining rules. Based on the information available in the scene graph, the system can then give appropriate answers: "I see a purse on the chair. Is that what you were looking for?" or "there is a water cooler by the wall" – see for example Figure 3.

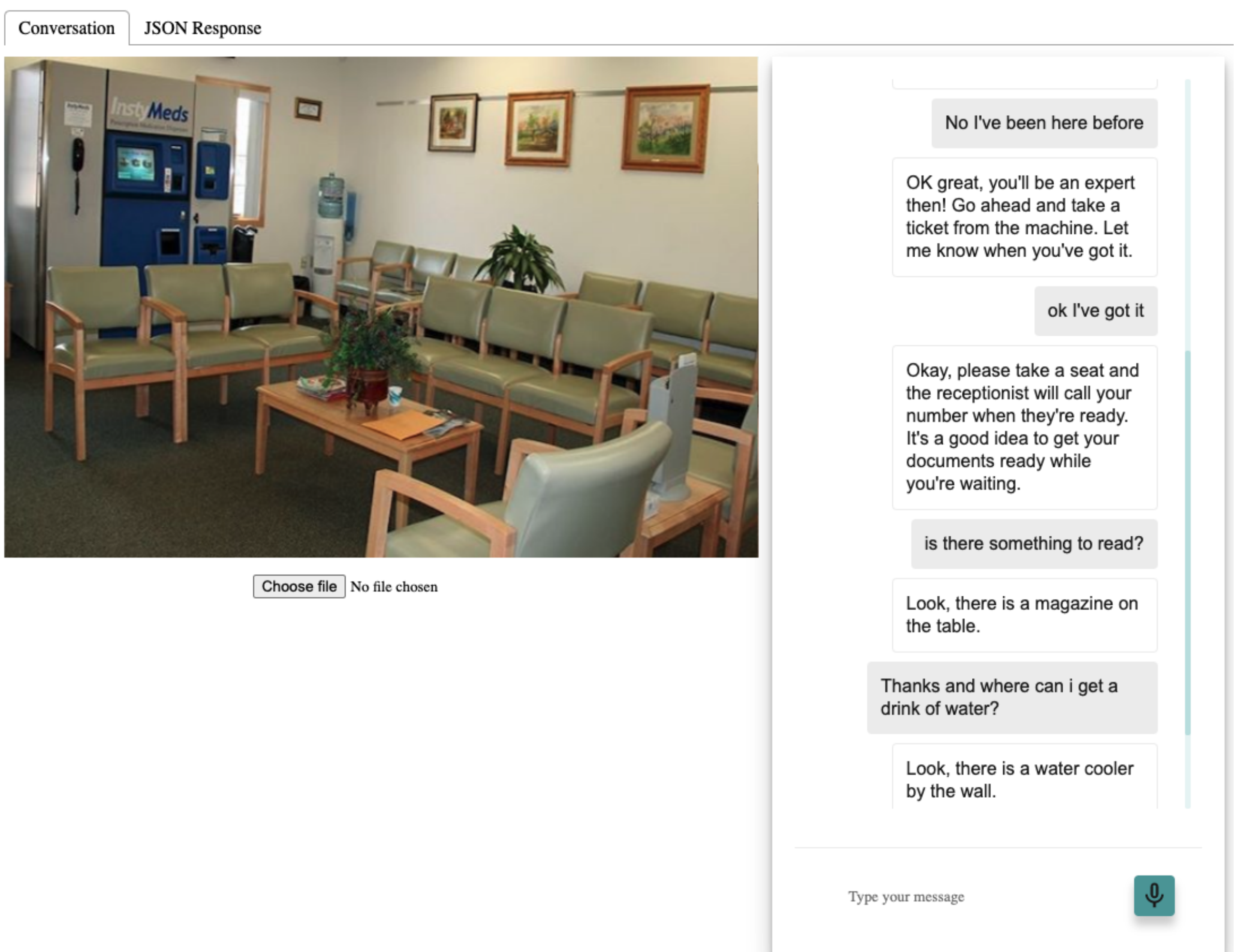

Figure 3: Web interface for the Initial Conversational System, showing visual dialogue

5 Example Dialogues

Given the above system architecture and a list of domain-specific bots, the initial SPRING system is able to to deal with a wide range of natural, daily conversations with end users that vary in complexity; from simple chitchat and quiz games, to domain-specific conversations and complex visual dialogue - see examples 1 to 4.

- Example 1 shows an example dialogue, consisting of chitchat and receptionist conversations, by Persona Bot, Direction Bot as well as Reception Bot. It aims at helping the patient prepare their appointment with the Doctor or find the direction to their destination.
- Example 2 presents a domain-specific conversation regarding coronavirus. The example illustrates the corona bot's three main capabilities: providing basic facts on Covid-19 e.g. on prevention and key symptoms; myth-busting and mental health support.
- Example 3 is an example of Visual Dialogue, in which the robot helps the user find his "blue" jacket ....
- Example 4 presents a quiz game example in the field of Covid-19, operated by the Quiz bot. The robot is able to invite the user to play a game when the Robot detects that someone has waiting for a long time.

U: Hello
R: [reception_bot] **Hi there. How can I help?**
U: I'm Jeanne XX. I have an appointment with Dr X at 8.30 am.
R: [reception_bot] **Pleased to meet you, Jeanne. Is this your first visit?**
U: Yes it is.
R: [reception_bot] **Welcome to Broca, we'll be sure to look after you. First, you need to take a ticket from the machine here.** ***[Robot points to ticket machine]***
U: Right, OK.
[...]
R: [reception_bot] **Do you have the ticket?**
U: Yes, number 432.
R: [reception_bot] **OK, let me find you a seat...** ***[Robot looks round to identify suitable chair]***
R: [task_bot] **I can see an empty chair over there.** ***[Robot points in the direction of the empty chair]***
R: [reception_bot] **Take a seat and the receptionist will call your number when they're ready.**
U: OK thank you.
R: [persona_bot] **You're welcome. Hopefully, we'll get a chance to chat later!**
[...]
U: Actually, I need to go to the bathroom. Do you know where they are?
R: [direction_bot] **Yes of course. They are opposite the dining room, through the double doors from the main waiting room. Go through the double doors past the nursing station and they will be on your right.** ***[Robot points to the direction of the bathroom]***
U: Great, thank you.
R: [persona_bot] **No problem. Let me know if you need any more help or you'd like to have a chat.**

Table 1: Dialogue Example 1: Receptionist Conversation & Chit-chat

6 Outputs and Future Work

The initial High-level Task Planner and Conversational System Prototype have been implemented and integrated with the ARI robot system, in a simulated environment of a hospital's waiting room [Part et al., 2021] – see the video demo at HRI 2021: `https://youtu.be/eY_BNxwrlPg`. These components will later on be enhanced in tasks T5.2 and T5.3, and connected to the non-verbal behaviour manager of WP6.

In the simulated scenario, an individual, e.g., an elderly person, is looking for a place to sit, the person asks the robot to find a *chair*. This prompts the planner to simultaneously send a request to the vision system and to execute a turn-in-place action looking for the object specified by the user through the previous dialogue act. Once the vision system finds the target object, it relays this information to the interaction system which interrupts the physical action to execute a dialogue action informing the user of the outcome. If on the other hand, no object was found after the physical action has completed, the interaction system interrupts the vision system and executes a dialogue action to report this to the user producing an appropriate response.

For this demonstration, the planner and the conversational system have been integrated with an initial vision system with the other modules in the robot architecture. These efforts to develop visual semantic dialogue will continue in future tasks integrating WP2 and WP5.

U: How can I stay safe? R: [corona_bot] To protect yourself and others against Covid-19: wash your hands frequently; avoid touching yourface and stay 2 metres away from others. Don't forget I can talk about quarantine and how long the virus takes to incubate. I'm here to help! U: OK, how long does it take from exposure to getting sick? R: [corona_bot] That's a good question. According to the WHO, the time between exposure to Covid-19 and when symptoms start is commonly around 5 to 6 days but can range from 1 to 14 days. Now, what do you want to ask? U: What are the main symptoms to look out for? R: [corona_bot] The most common symptoms listed by the WHO are: fever; dry cough, and fatigue. Do you want to hear other, less common, symptoms? U: No thanks. R: [corona_bot] OK. I've heard a lot of rumours around Covid-19 - about garlic, nasal sprays, alcohol, bleach - try me with one and I'll let you know if it's true. U: What's the rumour about garlic? R: [corona_bot] Garlic is healthy food - and delicious if you don't mind having bad breath! But the WHO says there's no evidence that eating it protects you from coronavirus. What do you want to talk about now? U: When's it all going to end? R: [corona_bot] Alas, no-one knows how long the pandemic will last. The licensing of vaccines is a big first step towards resuming normal life, but it will take many months to vaccinate everyone who needs it, and it's not yet known how long protection will last. How else can I help you cope with what's going on? U: It's all getting me down. R: [corona_bot] It's a difficult time, but there are lots of things you can do to protect your mental health. If you're concerned, ring the Samaritans free on 116 123. Or I have some tips from the U.K. charity MIND on how to look after yourself. Do you want to hear them? U: Yes, please. [continue]

Table 2: Dialogue Example 2: Corona Chat - The user ask for information about Covid-19

R: [reception_bot] Hi there, can I help you with something? U: Can you see my cup? I forgot it here. R: [task_bot] OK, let me see. *[Robot looks around and searches for a cup]* R: I can see a cup on the table next to the plant. U: Oh yes that's it! Thank you! R: [persona_bot] Happy to be of service! U: Excuse me, where can i get a drink of water? R: [task_bot] *[Robot looks around and searches for a water cooler]* There is a water cooler by the wall. U: Aha, thanks again. R: [persona_bot] Happy to be of service!

Table 3: Dialogue Example 3: Visual Dialog - helping users find their belongings

R: [visual_engage_bot] Hello there. I noticed you've been waiting for a while. Would you like to play a game to pass the time? U: Not really, no. R: [quiz_bot] Are you sure? It's easy to play and a lot of fun. U: Alright, go on then. R: [quiz_bot] Great, let's play. please remember, whenever you want to leave the game, just say end game. now here is the first question, Is the virus mutating rapidly? U: Yes R: [quiz_bot] I think you said true. actually, it is false, the novel coronavirus is periodically changing, or mutating, in form. next question, Can you always tell if someone has COVID-19? U: False. R: [quiz_bot] You are right. next question, Are people living with HIV always more at risk? U: Um that one might be true. [....] U: end game. R: [quiz_bot] Thanks for playing, That was fun. Please let me know if you need any help.

Table 4: Dialogue Example 4: Quiz Game - engaging users while they feel bored (Robot approaches person seated alone for some time.)

In addition to the integration with the simulated robot, the system has been interfaced using a web-based chat interface to allow testing and validation. This was done to mitigate the impact of the COVID-19 pandemic on the project's progress, as constraints due to the pandemic still impede conducting experiments in our research laboratories. This system is ready for the first integration cycle T7.3, and will be used for the data collection task of T1.2 for training data-driven modules and to support the data generation of multi-party situated interactions module for T6.2.

For the initial high-level planner a social decision-making component will be developed and integrated with the behavioural manager of T6.3, in order to make decisions such as which person to interact with, where to go, and what task to execute at what time, etc. The high-level task planner will decide which is/are the forthcoming task/s to be performed by the robot, this will be based on the specified SPRING use-cases and on the social belief state [García et al., 2020], combining the different network streams from various modules, e.g., semantic mapping and localization (WP2), visual/audio features (WP3), human behaviour understanding (WP4), current dialogue and interaction status (WP5), and robot behaviours (WP6).

The initial conversational system will be extended for multi-party interactions.

References

SPRING-WP5-Repository. SPRING-WP5-Repository. URL https://gitlab.inria.fr/spring/wp5_spoken_conversations.

Ioannis Papaioannou, Amanda Cercas Curry, Jose L. Part, Igor Shalyminov, Xinnuo Xu, Yanchao Yu, Ondřej Dušek, Verena Rieser, et al. Alana: Social Dialogue using an Ensemble Model and a Ranker trained on User Feedback. *Alexa Prize Proceedings*, 2017a. URL https://s3.amazonaws.com/alexaprize/2017/technical-article/alana.pdf.

Ioannis Papaioannou, Amanda Cercas Curry, Jose L. Part, Igor Shalyminov, Xu Xinnuo, Yanchao Yu, Ondřej Dušek, Verena Rieser, and Oliver Lemon. An Ensemble Model with Ranking for Social Dialogue. In *Workshop on Conversational AI at NeurIPS*, 2017b. URL http://arxiv.org/abs/1712.07558.

Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xinnuo Xu, Ondřej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser, et al. Alana v2: Entertaining and Informative Open-Domain Social Dialogue using Ontologies and Entity Linking. *Alexa Prize Proceedings*, 2018. URL https://s3.amazonaws.com/dex-microsites-prod/alexaprize/2018/papers/Alana.pdf.

Mary Ellen Foster, Bart Craenen, Amol Deshmukh, Oliver Lemon, Emanuele Bastianelli, Christian Dondrup, Ioannis Papaioannou, Andrea Vanzo, Jean-Marc Odobez, Olivier Canévet, Yuanzhouhan Cao, Weipeng He, Angel Martínez-González, Petr Motlicek, Rémy Siegfried, Rachid Alami, Kathleen Belhassein, Guilhem Buisan, Aurélie Clodic, Amandine Mayima, Yoan Sallami, Guillaume Sarthou, Phani-Teja Singamaneni, Jules Waldhart, Alexandre Mazel, Maxime Caniot, Marketta Niemelä, Päivi Heikkilä, Hanna Lammi, and Antti Tammela. MuMMER: Socially Intelligent Human-Robot Interaction in Public Spaces, 2019. URL http://arxiv.org/abs/1909.06749.

Christian Dondrup, Ioannis Papaioannou, and Oliver Lemon. Petri Net Machines for Human-Agent Interaction, 2019. URL http://arxiv.org/abs/1909.06174.

Oliver Lemon, Alexander Gruenstein, Alexis Battle, and Stanley Peters. Multi-Tasking and Collaborative Activities in Dialogue Systems. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*, pages 113–124, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W02-0216.pdf.

Ioannis Papaioannou, Christian Dondrup, and Oliver Lemon. Human-Robot Interaction Requires More Than Slot Filling - Multi-Threaded Dialogue for Collaborative Tasks and Social Conversation. In *Proceedings of the FAIM/ISCA Workshop on Artificial Intelligence for Multimodal Human Robot Interaction*, pages 61–64, 2018. URL https://www.isca-speech.org/archive/AI-MHRI_2018/pdfs/10.pdf.

Igor Shalyminov, Ondřej Dušek, and Oliver Lemon. Neural Response Ranking for Social Conversation: A Data-Efficient Approach. In *Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI at EMNLP*, pages 1–8, Brussels, Belgium, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-5701.

Jose L. Part, Daniel Hernández García, Yanchao Yu, Nancie Gunson, Christian Dondrup, and Oliver Lemon. Towards Visual Dialogue for Human-Robot Interaction. In *Companion Proceedings of the 16th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 670–672, Boulder, CO, USA, March 2021. URL https://doi.org/10.1145/3434074.3447278.

Daniel Hernández García, Yanchao Yu, Weronika Sieinska, Jose L. Part, Nancie Gunson, Oliver Lemon, and Christian Dondrup. Explainable representations of the social state: A model for social human-robot interactions. *CoRR*, abs/2010.04570, 2020. URL https://arxiv.org/abs/2010.04570.