Towards efficient human-robot cooperation for socially-aware robot navigation in human-populated environments: the SNAPE framework

A. Vega-Magro[1], R. Gondkar[2], L.J. Manso[3] and P. Núñez[1]

***Abstract*— It is widely accepted that in the future, robots will cooperate with humans in everyday tasks. Robots interacting with humans will require social awareness when performing their tasks which will require navigation. While navigating, robots should aim to avoid distressing people in order to maximize their chance of social acceptance. For instance, avoiding getting too close to people or disrupting interactions. Most research approaches these problems by planning socially accepted paths, however, in everyday situations, there are many examples where a simple path planner cannot solve all of the predicted robots' navigation problems. For instance, requesting permission to interrupt a conversation if an alternative path cannot be determined requires deliberative skills. This article presents the Social Navigation framework for Autonomous robots in Populated Environments (SNAPE), where different software agents are integrated within a robotics cognitive architecture. SNAPE addresses action planning aimed at social-awareness navigation in realistic situations: it plans socially accepted paths and conversations to negotiate its trajectory to reach targets. In this article, the framework is evaluated in different use-cases where the robot, during its navigation, has to interact with different people in order to reach its goal. The results show that participants report that the robot's behavior was realistic and human-like.**

I. Introduction

The navigation of autonomous robots in indoor environments has been one of the main challenges of robotics in recent decades. Navigation involves robot skills such as path planning and efficiently moving through the environment. One of the goals of socially aware robotics regarding navigation is to replicate human behavior as closely as possible.

Socially aware robots, especially those able to navigate, must be capable of detecting people, their locations and their interactions. This information provides the basis for social aware navigation, which uses concepts such as proxemics theory or object space affordances to identify the personal spaces of interaction that the robot should avoid while navigating. However, there are situations that prevent robots from reaching their destinations without requiring external cooperation. For instance, the situations described in Fig. 1. In Fig.1a, two people are in a conversation and the robot plans a route that requires to pass through the space that these two people share. In Fig. 1b, a person in a corridor interrupts the path of the robot. What a social-aware robot should do in these situations is still an open question. Once

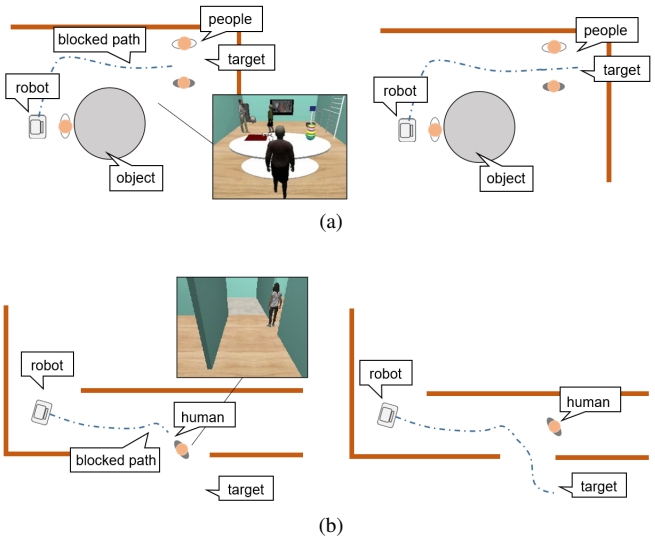


Fig. 1: a) (left) two people interacting interrupt the path planned by the robot; (right) after HRI, the robot can continue its navigation; b) (left) path is blocked due to a human next to the door; (right) after HRI the robot's path is unblocked

robots perceive and interpret the social contexts considering social conventions, they must plan a sequence of actions (*e.g.*, approach the person, call their attention, or start a social dialogue to ask for cooperation).

To the authors' knowledge, this study is the first work that addresses all the planning of actions aimed at the social-awareness navigation of robots in realistic situations including cooperation. This is the **main contribution** of the article: a Social Navigation framework for Autonomous Robots in Populated Environments (SNAPE), which manages the actions to be carried out, the dialogue flow, planning of the robot's path, and the perception environment for social awareness navigation. The proposed framework answers the hypothesis that robots need social behaviors and cooperation from humans to navigate socially. This cooperation is achieved using different layers and levels of planning in SNAPE. The robotic architecture used is CORTEX [1], the evolution of AGM [2] where multiple software agents are coordinated through a shared graph-like world model. The planning domain is defined using AGGL, a planning domain definition language created for AGM. The specific dialogues are established through the *RASA chatbot API* [3].

[1] A. Vega-Magro and P. Núñez are members of the Robotics and Artificial Vision Lab. RoboLab Group, University of Extremadura, Spain. pnuntru@unex.es

[2] R. Gondkar is with the Pune Institute of Computer Technology, Pune University, India. rishigondkar@gmail.com

[3] L.J. Manso is with the College of Engineering and Physical Sciences, Aston University, Birmingham, UK. l.manso@aston.ac.uk

II. Related work

To design socially-aware robot navigation that meets user needs, it is critical to take into account social considerations that influence the robot motion in the environment. In a realistic scenario, these social considerations attend to the way the robot navigates, but also respect a comfortable human interaction space [4]-[8]. These works are based on *social mapping*, a concept that goes beyond metric and semantic mapping. Other theories use learning-based methods in which the robot learns its social behaviors by observing how humans navigate [9], [10].

It is also necessary to take into account the possibility of planning social behaviors, such as approaching people to talk to them or requesting permission to interrupt conversations. The previous works present the same fundamental limitation when the robot navigates in environments with humans: the planned path has the potential to be interrupted by people or their interactions. Planning HRI for social navigation is a topic of growing interest, although there are currently few works on the subject. In [11], two scenarios are used to introduce the concept: a person blocking a path and two interacting people blocking the path. A navigation planner that takes into account HRI for some of the sub-problems of social path planning is proposed in [12]. In [8] the authors propose a framework for social navigation using modules for planning or conversation. However, the dialogues in these and other similar works in the literature are basic or non-existent, and disregard social acceptance.

HRI for dialogue is used in different scenarios, such as modeling and planning [13]. HRI for navigation is done almost exclusively to send orders of movement to the robot. In [14], the authors defined the corpus focusing on this premise, moving a robot away from the human. A human operator sends basic commands to the robot in a similar work [15]. In works like [16], there is also a dialogue, however it is not part of any navigation architecture. The dialogue to cooperate within a socially-aware navigation framework, initiated by the own robot, is novel. The authors in [17] presented the original idea, which is limited to simple interactions to enable robots to continue their motion. The proposed framework extends this idea by enriching conversations and defining the corpus of collaborative dialogues whose final goal is to improve cooperation for social-awareness navigation.

III. Social Navigation framework for Autonomous robots in Populated Environments

An overview of the proposed framework is described in Fig. 2. The SNAPE framework is built on five levels: 1) perception layer; 2) social layer; 3) navigation layer; 4) HRI interaction layer, and 5) planning layer. All these layers are associated with independent software agents of the CORTEX architecture [1], which shares information of the world according to the Deep State Representation (DSR). The use of the cognitive architecture CORTEX and the DSR allows the improvement and extension of functionalities of each agent keeping the framework structure, *i:e:;* it is not limited

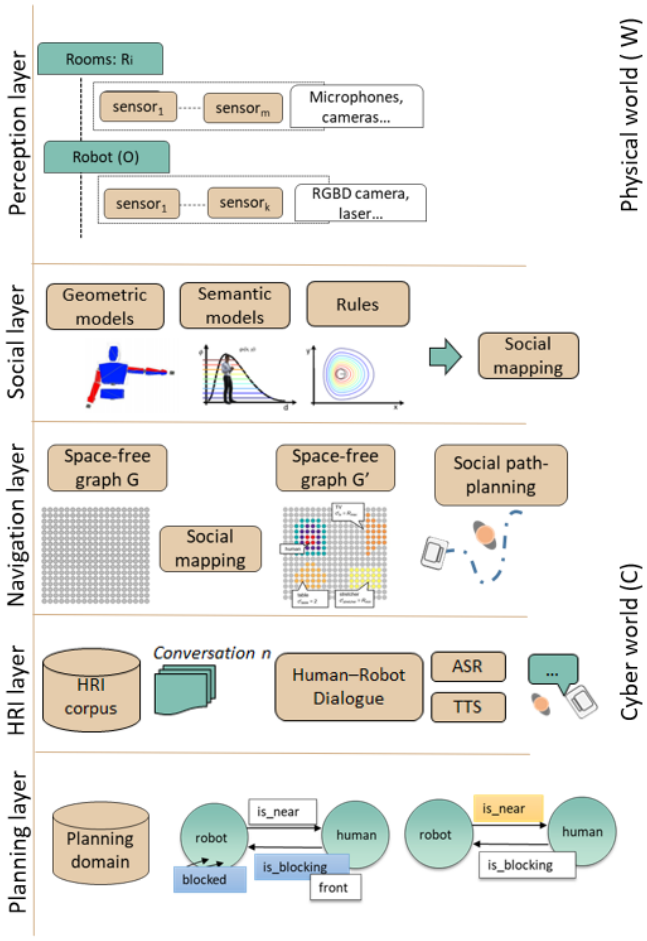


Fig. 2: Overview of the proposed SNAPE framework

to the use of an algorithm or other, but, once defined the functionality of each layer, it can be substituted by specific algorithms and agents. The SNAPE framework covers all the spectrum: from the perception of the environment to behavior planning. SNAPE includes planning at different levels: at the level of dialogue, task-planning and behavior.

A. Perception layer

The lowest level in the architecture represents the robot's need to acquire enough information from the environment to make socially acceptable decisions. Social navigation depends on factors related to the robot's surroundings: the position of people and objects, dimensions of the environment, or the hour of the day. Robots that intends to be socially accepted have to adjust their navigation to these elements and be capable of detecting them.

The perception of the environment is made through the robot's sensor readings, which can be integrated into a smart ecosystem equipped with sensor networks. The perception layer constitutes the physical world inside the cyber-physical ecosystem [18], and it is composed of cameras and microphones.

The perception layer uses the models described in [7], [18]. On the one hand, in the case of a person *i* in the

environment, $h_i = (x; y;)_i$, is described by its position $(x; y)$ and orientation , which are tracked by a specific agent. On the other hand, an object j, o_j, is described by its pose $p_{o_j} = (x; y;)_j$. The detection of objects and people in the environment is out of the scope of the paper and are assumed to be detected by the agents in CORTEX. Finally, $H_n = \{h_1; h_2 ::: h_n\}$ and $O_m = \{o_1; o_2 ::: o_m\}$ are the sets of n and m humans and objects detected by the perception layer. This information is updated in the DSR to ensure that all agents in the architecture share the same knowledge from the robot's surroundings.

B. Social layer

The second layer of the architecture represents the robot's need to be socially aware. Social awareness depends mainly on socially accepted factors. A social robot, for instance, should respect social distances, should not interrupt a conversation or request permission if possible. In the SNAPE framework, several software agents take part in the definition of the social mapping [18].

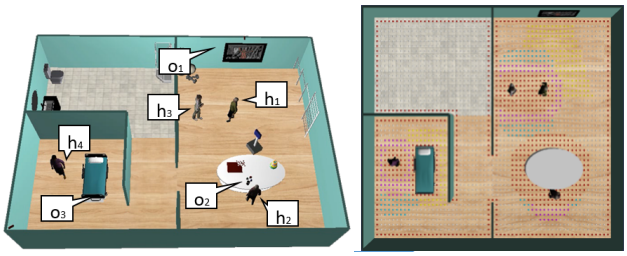
- *Social mapping: populated environments.* Given H_n, the set of people detected in the perception layer, the interaction spaces of each individual h_i are modeled as the asymmetric 2-D Gaussian curves $g_i(x; y)$ [7]:

$$g_{h_i}(x, y) = e^{-(k_1(x-x_i)^2 + k_2(x-x_i)(y-y_i) + k_3(y-y_i)^2)} \quad (1)$$

  being k_1, k_2 and k_3 a set of coefficients which are dependent on the orientation $_i$. This Gaussian function emphasizes the region in front of the person, as defined by the theory of proxemics. Once people have been detected, the algorithm clusters interacting people in the environment according to their distances by performing a Gaussian Mixture, as described in [7]. The personal space function $g_i(h)$ of each individual h_i in the environment is totalled and a global interaction space $G(h)$ is built.

- *Social mapping: Space Affordances and Activity Spaces.* Given O_m, each object $o_k \in O_M$ also stores the interaction space i_{o_k} as an attribute, which is associated to the space required to interact with this object. These spaces have been modeled depending of the shape of the object and the way that people interact with: i) TV or poster similar shapes; ii) rectangle shapes (*e:g:*, beds or tables); and iii) circular shape objects (*e:g:*, tables). Interaction spaces are added only if the person is interacting with the object.

These interaction spaces are updated in the DSR through specific links and nodes. Fig.3a shows a 3D view of a simulated scenario with four people and three objects, $H_4 = \{h_1; h_2; h_3; h_4\}$, $O_m = \{o_1; o_2; o_3\}$. Fig.3b shows the result of applying the social layer in Fig.3a (*i:e:;* social mapping), where the social interaction spaces have been represented in different colors.



(a) (b)

Fig. 3: a) People and objects in a simulated environment; b) social mapping built from the social layer of the framework.

C. Navigation layer

The robot's environment is represented by a uniform graph composed of obstacle-free nodes with a fixed finite traversal cost, and non-free nodes, which have an infinite one. The SNAPE framework modifies the costs according to the social map [7]. This final graph is used to estimate the optimal social path using the classical Dijkstra's algorithm.

- *Graph-based grid mapping:* Space is represented by a graph $G(N; E)$ of n nodes, regularly distributed in the environment. Each node n_i has two parameters: availability, a_n, and cost, c_n. The availability of a node is a Boolean variable whose value is 1 if the space is free, 0 otherwise. The cost, c_i, indicates the traversal cost of a node, *i:e:;* what it takes for the robot to visit that node (high values of c_i indicates that the robot should avoid this path). Initially, all nodes have the same cost of 1.
- *Social graph-based grid mapping:* The node parameters a_n and c_n of the free space graph are modified according to the areas defined in the social map: firstly, to include those associated with the interaction between one person and another (or groups of people), and secondly, to include those associated with the affordance spaces of objects.
- *Socially-acceptable path-planning:* Dijkstra is used to determine the shortest path between an initial position and a target to which the robot must travel. Given a source node, the algorithm calculates the cost to the target node, taking into account the cost of the nodes. The cost of a path is the sum of the cost of the nodes it is composed of.

D. Human-Robot interaction layer

The fourth level of the SNAPE framework represents the need for the socially-aware robot to begin specific interactions that arise during navigation. In this article, the dialogue has been developed for three particular situations. The first represents the dialogue when a single person blocks the path and must move for the robot to pass through. In the second dialogue, the robot asks permission to interrupt a conversation between two or more people. A third intermediate dialogue is generated to get the person's interest, in case the

robot wants to initiate an interaction with the person and they are not face-to-face.

This layer includes the following subsystems: i) Natural Language Understander (NLU, *i.e.*, translates natural language human utterances from the individual side to a formal semantic representation); ii) Natural Language Generator (NLG, *i.e.*, translates statements in formal semantic representation from the robot side to natural language utterances); and (iii) Dialogue State Tracker (DST, *i.e.*, is responsible for maintaining the state and flow of the dialogue, choosing the best conversation from the dialogue corpus.

This corpus is created from the "Wizard-of-Oz" methodology. Under this approach, the participant believes that they are blocking the robot's path, and thus, the robot asks for cooperation and starts the dialogue. However, a human is performing the NLU function, translating the participant's utterances from natural language to semantics. This same human is in charge of the NLG system, directly typing the responses. All utterances are recorded and analyzed, annotating intents, actions, and conversations.

The flow of the dialogue is established through the RASA framework, a conversation system based on *Intents*, *Entities*, *Stories* and *Actions*. The RASA NLU analyses and places each phrase in one *Intent* depending on the keywords previously defined in the study. For instance, if a person in the robots surrounding says "Good morning" the NLU returns the "greeting" *Intent*, or if the person says "Yes, I surely will" the NLU returns the "affirmative" *Intent*. According to the *Intent* recognized, the Rasa core, which is trained on the *Stories* created using the 'Wizard-of-Oz' approach, predicts the next *action* of the robot. For instance, when the "negative" *Intent* is active, RASA Core predicts the *action* "utter_frustrated" and then the robot is ready to listen on that topic after its execution. The current situation in the real world is stored in the *Entities*, which are used to direct the flow of the dialogue accordingly. Within the established corpus in the SNAPE framework, there are five classes of conversation *Intents* defined: greeting, affirmative, negative, repeat, and conversation.

E. Planning layer

The fifth and last layer of the architecture represents the need for the robot to plan specific actions to carry out the navigation to the target. Planning HRI for navigation tasks entails defining the elements of the planning problem: an initial world model, a mission, and a set of actions (*i.e.*, the planning domain). In the SNAPE framework, planning is performed with the symbolic information in the DSR, using the nodes of the representation as symbols and the edges of the graph as predicates [11]. Fig. 4 illustrates the shared representation associated to Fig. 3a. As shown in Fig. 4, CORTEX uses different types of symbols and edges, however only symbols are used in the planning domain: *human*, *robot*, *objects* and *room*. Similarly, the set of edges are limited in the planning domain.

This paper is focused on those cases where only a *robot* is located in the model, but where several people, objects,

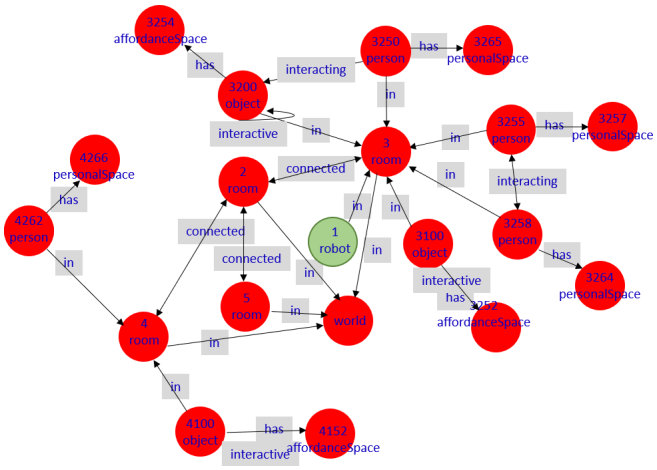


Fig. 4: Deep State Representation of the world shown in Fig. 3a.

and rooms are possible. The planning domain in the SNAPE framework defines all the rules needed to be socially accepted, among others: to navigate without disturbing people, approach a person or group of people, get their attention, and initiate interaction. The planning rules are described through AGGL [2], and thus, they are defined as pattern pairs, in the same way as string grammar rules: each rule states that the pattern on the left-hand side can be replaced with the pattern on the right-hand side. Fig. 5 shows the set of rules described in the SNAPE framework[1]. In the figure, the blue color indicates the nodes and edges on the left-hand side, in yellow, the elements that will be on the right-hand side, and in green and white, the nodes and edges, respectively, that do not change in the rule. For example, the *changeRoom* action in Fig. 5 shows to the robot *in* a room in the left-hand side (initial state) and, on the right-hand side, after applying the rule, the robot should be *in* another room. Both rooms are accessible and the robot is not blocked. The set of rules determines the robot's pose when it approaches people to engage the dialogue, and besides, it defines the case in which the person does not retreat after the conversation.

F. Case study

To better communicate how the proposed framework can guide the design of socially aware navigation behavior in robots, two examples are presented: an only person blocks the path, and two people interacting with each other block the robot's path. In both situations, the robot navigates in a populated environment (goToRoom action). The perception layer detects people and objects. Then, the social map modifies the values of the graph $G(N; E)$, and the robot plans its social path. If this path is blocked during navigation, new missions are generated: approaching a person (*e.g.;* goToPerson action), drawing its attention with a specific dialogue (*e.g.;* takeTheAttention action), asking permission to pass (*e.g.*, askForPermission action). The management of

[1]These rules, as well as the rest of the planning domain, are available in https://github.com/robocomp/robocomp.

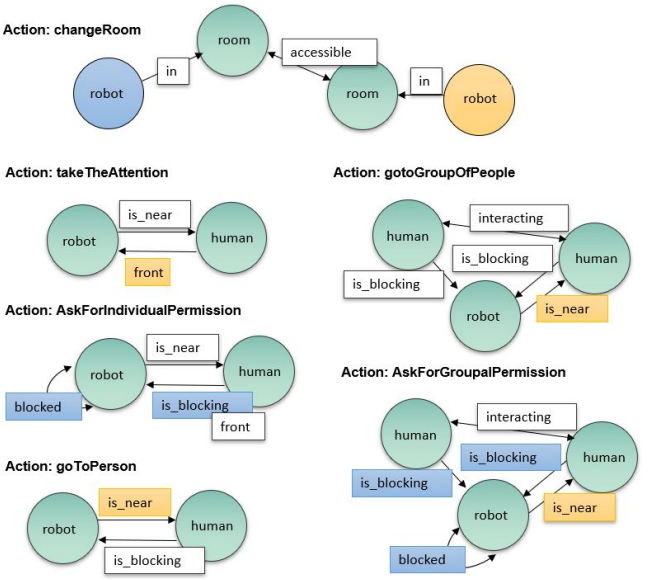


Fig. 5: Set of planning rules defined in the SNAPE framework. See the text for more details.

the dialogue and actions is the responsibility of the HRI and planning layer, respectively.

IV. Experimental Results

A set of simulated scenarios were used to validate the results of the SNAPE framework and the proposed corpus. The algorithms have been developed in C++ and the tests have been performed in a PC with an Intel Core i5 processor with 4Gb RAM and Ubuntu 18.10. The framework runs in real-time. A total of 20 participants evaluated the SNAPE framework and the dialogues by interacting with the robot in simulated scenarios.

This simulated scenario is a $65m^2$ apartment with two rooms, a corridor, and one bathroom, in which different RGBD cameras are installed. The social robot is an omnidirectional base equipped with an RGBD camera. At the beginning of each experiment, people were randomly placed in the environment. Some of them blocked the robot's path in the corridor, while others talked in a vis-a-vis formation blocking the path. During the robot's navigation, participants acquired the role of the human that blocked the path. All people moved through a Graphical User Interface controlled by the participants. Robot and participant interacted, where the dialogue corpus was used for generating natural language utterances. To avoid a biased evaluation that can occur when different TTS/ASR algorithms are used, the dialogue was carried out directly by sending text messages on the GUI. Each participant decided the behavior of the human in the simulated scenario (*e.g.*, choosing whether to let the robot cross or not). To validate the SNAPE framework and assess the satisfaction of the humans regarding the robot's behavior participants completed a Likert scale-based questionnaire. The results of the questionnaire, including some of the questions, are shown in Table I.

Question	avg. (σ)
The robot navigates in a similar way to the human	4.61 (0.44)
The robot correctly approaches the person to start a conversation	4.27 (0.40)
The robot correctly proves its intention of wanting to start the conversation	4.67 (0.38)
The robot asks for cooperation to continue its navigation kindly	4.1 (0.65)
The robot responds appropriately during the conversation	4.25 (0.32)
The structure of the dialogues is appropriate to ask cooperation	4.05 (0.54)
The robot understands the social context and the interaction	4.44 (0.42)
The robot shows socially accepted behavior	4.65 (0.32)

TABLE I: 20 participants used a Liker scale-based questionnaire to evaluate the dialogue corpus and the framework.

Fig. 6 shows the evolution of the DSR during robot's navigation in one of the many tests. Fig. 6a illustrates four time instant labeled from 1 to 4. Fig.6b shows the state of the DSR in those same instants of time. The current state of the DSR allows the planning layer to generate the appropriate actions, which are also presented in Fig.6b[2]

As shown in Table I, most of the participants agree that the robot's behavior for the study case is socially appropriate and friendly. Besides, most of them also agree that the robot understands the social contexts and interactions in all the scenarios. One of the main ideas after studying this Liker scale-based questionnaire is that the dialogue corpus presented in this paper achieves its function in the architecture correctly. In general, all participants agree that the dialogues improves the cooperation for navigating. Another main conclusion is that the SNAPE framework allows a more realistic socially-awareness navigation.

V. Conclusions

Human-aware robot navigation in populated environments is a complex problem that is currently unresolved. The situations that prevent a robot from reaching its end pose are extensive, and in some of them, robots must ask for some cooperation from the people around them. In this work, the framework SNAPE is described, including but not exclusive to a human-aware navigation architecture that integrates robot's skills such as the perception of its surrounding, the definition of the social map of the scene, the human-aware path planning and navigating, the planning of dialogues to solve situations where the robot cannot continue navigating, and the planning of high-level social actions.

The SNAPE framework provides the basis of social-awareness navigation. Conceptually, it is divided into layers with specific functionalities which are easily adaptable to other platforms. The framework has been evaluated in simulated environments, and the results demonstrate that the robot navigates and interacts following social conventions. Finally, a dialogue corpus has been created for the real scenarios covered in this paper. All software is open-source and available, including these dialogues.

ACKNOWLEDGMENT

This work has been partially supported by the Extremaduran Goverment project GR15120, IB18056, and by the

[2]Readers can find a video of some of the tests in `https://www.youtube.com/watch?v=KiWC2M9qjmY`