Simulating Human-Robot Interactions for Dialogue Strategy Learning

Grégoire Milliez, Emmanuel Ferreira, Michelangelo Fiore, Rachid Alami, Fabrice Lefèvre

Simulating human-robot interactions for dialogue strategy learning

Grégoire Milliez[1⋆], Emmanuel Ferreira[2⋆⋆], Michelangelo Fiore[1⋆⋆⋆], Rachid Alami[1†], and Fabrice Lefèvre[2‡]

[1] CNRS, LAAS, 7 avenue du colonel Roche, F-31077 Toulouse, France / Université de Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31077 Toulouse, France
[2] LIA, Université d'Avignon BP1228 - 84911 Avignon Cedex 9, Avignon, France

Abstract. Many robotic projects use simulation as a faster and easier way to develop, evaluate and validate software components compared with on-board real world settings. In the human-robot interaction field, some recent works have attempted to integrate humans in the simulation loop. In this paper we investigate how such kind of robotic simulation software can be used to provide a dynamic and interactive environment to both collect a multimodal situated dialogue corpus and to perform an efficient reinforcement learning-based dialogue management optimisation procedure. Our proposition is illustrated by a preliminary experiment involving real users in a Pick-Place-Carry task for which encouraging results are obtained.

1 Introduction

Simulation softwares are highly needed in robotic projects. By using simulators, roboticists can evaluate and validate their works on the chosen level of abstraction in a sandbox that limits the risk-taking. In this way, projects relying on high level computation (e.g. interaction, dialogue, supervision), can use a simulator to abstract lower levels (e.g. navigation, image processing, localisation) and avoid their related issues to interfere during the system evaluation. Furthermore, the simulation setup can also be useful to assess parts of the development before any attempt of costly integration on the on-board robotic platform.

In the present study, we specifically focus on Human-Robot Interaction (HRI). In the simulation context, two distinct solutions can be adopted to integrate humans in the loop: 1. modelling and implementing their behaviours and actions, and 2. dealing with tele-operation to control human avatars.

The first solution has the advantage of automatization and does not require manual manipulation. So, this solution is less time consuming and easier to run.

⋆ gregoire.milliez@laas.fr
⋆⋆ emmanuel.ferreira@univ-avignon.fr
⋆⋆⋆ michelangelo.fiore@laas.fr
† rachid.alami@laas.fr
‡ fabrice.lefevre@univ-avignon.fr

However, depending on the human features required, it may be really difficult to have a realistic human model. Humans are complex entities with reactions and behaviours nearly impossible to consistently synthesize. This solution is usually used for studies that do not involve the most complex sides of human behaviours, such as navigation or manipulation.

The second solution consits in having the simulated human controlled by a real human operator. By doing so, the complexity of the experimentation rises as an actual human is required to operate the human avatar. However, the avatar in the simulator will have a far more realistic behaviour. To do so, the simulator must have a realistic environment rendering and the human control must be natural and close enough to the real world.

HRI projects that focus on situated dialogue usually investigate tasks that seem to be within the scope or already implemented in HRI simulator, such as a Pick-Place-Carry scenario [1], robot bartender [2] or navigation tasks in a virtual environment [3]. Nevertheless, few works consider the simulation setup as the way to carry out situated dialogue corpus acquisition as well as a test-bed for an efficient online dialogue policy learning. Indeed, most of the previous works in situated dialogue for HRI resorted to a preliminary Wizard-of-Oz (WoZ) experiment, where a human remotely operates the robot [4,2,5]. However, the WoZ technique is both time consuming and an expensive method.

In this article we present how a robotic simulation software, in which the human is integrated, can help to train a dialogue system for realistic HRI from scratch. In Section 2 we explain how we simulated HRI scenarios with the open-source robotic simulator MORSE. In Section 3 we show how we used the simulator along with a robotic architecture, and finally in Section 4 we expose the integration with the dialogue system and give some testing results. In last Section 5, we discuss the outcome of this preliminary study and future work.

2 MORSE as HRI Simulator

2.1 Why MORSE for HRI?

In the robotic field, many simulators are available. We can name the Player/Stage/Gazebo suite [6], the integrated simulation platform OpenHRP [7], the cross-platform software architecture OpenRAVE [8] or even the commercial simulator V-REP [9]. However, only a few of them are very well suited to HRI. They generally limit human agent behaviours to relatively simple motions and interaction capacities which is one of the reasons why HRI simulations so far have been carried out in *tele-operation* settings, where only the robot and the environment, but not the human agent, are actually simulated. Robotic simulators USARSim [10] and MORSE [11,12] are both used in dozens of HRI studies due to their explicit support for controlling a human agent. However, the latter has several specific advantages that motivated our choice.

MORSE is an open-source simulator, with a very active community, that was developed specifically for robotic simulation. It supports a wide range of

middleware (e.g. ROS, YARP, pocolibs) as well as reliable implementations of realistic sensors and actuators which ease the integration on real robotic platforms afterwards. Moreover, MORSE offers an adaptable simulation setup by allowing virtual robots to interact with the virtual environment through both realistic sensors/actuators and higher level ones. Thereby, roboticists can control the related computation cost of low level data processing by exploiting high level outputs from unrealistic components. For example, MORSE provides both a vision camera and a semantic camera sensor. While the first camera provides a rough image (i.e. raw pixels) as output, the second one gives directly the names of the perceived objects and their positions in the scene. The latter sensor avoids practitioners to perform object recognition and localization processes when focusing on higher level issues.

Furthermore, MORSE relies on the Blender Game Engine, a real-time 3D runtime integrated to the open-source Blender modelling toolkit, for both advanced 3D (OpenGL shader) and physics simulation (based on the BULLET physics engine). This setup allows realistic rendering of complex environment and provides an immersive graphical user interface, which is a required feature for HRI modelling.

In MORSE, the human avatar can be controlled by a human operator or directly through external scripts as any other robot.

Fig. 1: Human avatar grabbing an object controlled by an operator (left image) and human in 3rd person perspective (right image).

In the first case, the operator controls the virtual human in an immersive way (see Figure 1) in terms of displacement, gaze, and interactions on the environment, such as object manipulation (e.g. grasp/released an object). To go even further in realistic human incorporation in the simulator, a motion capture actuator allows to control the human avatar directly by using an external device. So, a Kinect sensor collects human gestures and sends the posture data to move the human avatar accordingly. Furthermore, a Nintendo wiimote can jointly be used to manage its action (e.g. grasp/released an object).

In the second case, the avatar is programmatically controlled by using standard MORSE actuators. As an example, it is possible to use a waypoint actuator on the human to define a path he has to follow.

2.2 Scenario Implementation

In our scenario, a disabled human is in her apartment and has a robot to assist her to perform everyday life chores. The goal is to make the robot understand, by reasoning on human speech, gestures and the environment, human's requests concerning objects. Objects are limited here to graspable items such as books, DVDs and mugs, that have diverse colors and a unique identifier.

The PR2 robot is used in the simulator as it is our real platform at LAAS-CNRS. PR2 is already present in MORSE models, making it directly usable. We add a symbolic camera (MORSE semantic camera) sensor to the standard model so that it can perform object recognition and also a teleport actuator to move it to a designated position (while saving the time of the true displacement). We also add a human avatar with first person representation to have realistic inputs of speech and behaviour of human users. We use a virtual model of the physical environment in which the real robot will be tested (see Figure 2).

Fig. 2: Scenario environment in MORSE

At the start of the simulation, a script randomly positions objects in predefined areas (such as over kitchen table, living-room table, bedroom shelf etc.), called *manipulation areas*. This allows us to use different environment configurations without changing the initialization files (MORSE builder script).

2.3 Actions library

To get a more interactive and realistic simulation and also for the user to evaluate the fulfilment of her request (e.g. does the robot bring the appropriate object), we have developed a library of high-level and abstract actions that the robot will be able to perform.

The list of abstract actions is as followed:

– To explore the environment and bring an object to the human, the robot needs to be able to move to manipulation areas. To do so, we use the teleport actuator of MORSE. This actuator moves instantaneously the robot to a given place. We define a script function to move the robot to each manipulation area that has been defined. In this way the robot can go to each position to pick objects or explore an area to get some contextual information.
– The robot is able to scan a manipulation area. To make this action possible a symbolic camera is added to the robot on its head. We then move the head sequentially to scan the environment.
– The robot has to grab an object. To perform this action the grasp service of the PR2 is used. We specify the name of the object it has to grab and if the object is close to robot's hand it will be attached to it. In a similar way, we added a function to drop an object that takes as parameter the manipulation area where it should be dropped to. The robot will drop the object on top of the corresponding furniture.
– The last action is giving the object to the human. It consists in moving the robot to the human position and deploying the arm of the robot toward the human to give her the object. We simply use the robot armature actuator to control the robot's arm.

3 Integration with robotic system

3.1 SPARK for Spatial Reasoning

To achieve geometric reasoning and to get the environment through robot perception SPARK [13] (SPAtial Reasoning and Knowledge) was used on our robot. To do the same in our simulated environment we need to get data from MORSE. We briefly explain here how we linked SPARK with MORSE and then what is obtained from this integration.

SPARK gets three kinds of input: object identifier and position, human position and posture and robot position and posture.

To obtain the object position in SPARK, we use the semantic camera on the robot head. This sensor can export the position and name of objects in view field. This data is sent using a middleware and is then read by SPARK to position the object in its representation. Concerning human and robot, we attach a pose sensor to them and we export their armature configuration. In this way SPARK can read the position and posture of the robot and human through the middleware, requesting only a mapping to match the MORSE joint representation with the SPARK one.

SPARK uses robot perception data to build the environment as seen by the robot. It also computes geometrical facts such as topological description of object's position (`Book isOn table`), agents affordances (`Book is visibleBy Human`) and knowledge of agents (`Human hasKnownLocation Book`). These high level data will be used to enrich the dialogue context.

3.2 Robotic System

By using SPARK as a link in the system, we are able to use a full robotic architecture with MORSE. This architecture, shown in Figure 3, is composed of several modules:

- Supervision System: the component in charge of commanding the other components of the system in order to complete a task.
- HATP: the Human-Aware Task Planner [14], based on a Hierarchical Task Network (HTN) refinement [15]. HATP is able to produce plans for the robot actions as well as for the other participants (humans or robots).
- Collaboration Planners: this set of planners are used in joint actions such as handover to estimate the user intentions and selects an action to perform.
- SPARK: the Spatial Reasoning and Knowledge component, as explained in 3.1.
- Knowledge Base: the facts produced by the geometric and temporal reasoning component are stored in a central symbolic knowledge base. This base maintains a different model for each agent, allowing to represent divergent beliefs.
- Human Planners: a set of human aware motion, placement and manipulation planners [16].

Our system is able, using SPARK, to create different representations of the world for itself and for the other agents, which are then stored in the Knowledge Base. In this way the robot can take into account what each agent can see, reach and know when creating plans. Using HATP the robot can create a plan constituted by different execution streams for every present agent.

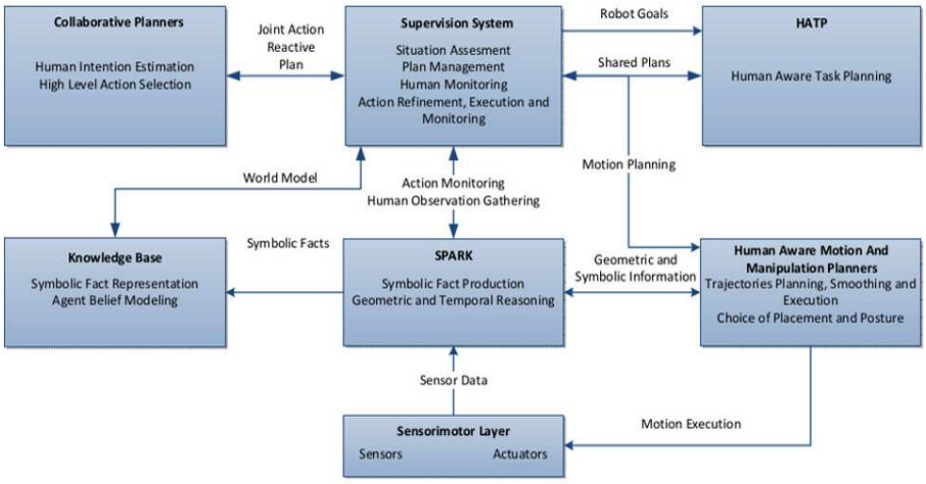

Fig. 3: Robotic system architecture

The interaction process is designed to be flexible. Users are able to issue commands to the robot but the robot is also able to plan on its own to execute

complex goals and to adapt its plan to user actions. Human actions are monitored using SPARK by creating Monitor Spheres associated to items deemed interesting in a given context. A Monitor Sphere is a spheric area surrounding a point that can be associated to different events, like the hand of a human entering into it. The system is explained in more details in [17].

4 Learning Dialogue Strategies

4.1 Dialogue Management

In this study the robot is dedicated to help a human achieving a specific object manipulation task. Thereby, multimodal dialogues are employed to solve ambiguities and to request missing information until task completion (i.e. full command execution) or failure (i.e. explicit user disengagement or wrong command execution). In this setup, the robot, more precisely the Dialogue Manager (DM), is responsible for taking appropriate multimodal dialogue decisions to fulfil the user's goal based on uncertain dialogue contexts.

To do so, the dialogue management problem is cast as a Partially Observable Markov Decision Process (POMDP). In this setup, the agent maintains a distribution over possible dialogue states, called the belief state in the literature, and interacting with its perceived environment using a dialogue policy learned by means of a Reinforcement Learning (RL) algorithm [18]. This mathematical framework has been successfully employed in the Spoken Dialogue System (SDS) field (e.g. [19,20,21]) as well as to manage dialogue in HRI context (e.g. [22,1]). Indeed, this framework explicitly handles parts of the inherent uncertainty of the information which the DM has to deal with (erroneous speech recognitions, misrecognized gestures, etc.).

Recent attempts in SDS have shown the possibility to learn a dialogue policy from scratch with a limited number (several hundreds) of interactions [23,24,25] and the potential benefit of this technique compared to the classical use of WoZ or to develop a well-calibrated user simulator [23]. Following the same idea, we employ a sample-efficient learning algorithm, namely the Kalman Temporal Differences (KTD) framework [26,25], which enables us to learn and adapt the robot behaviour in an online setup. That is while interacting with users. The main shortcoming of the chosen method consists in the very poor initial performances. However, solutions as those proposed in [27,28] can be easily adopted to alleviate this limitation.

Although objectively artificial, the presented robotic simulation platform provides a very interesting test-bed module for online dialogue learning. Indeed, a better control over the global experimental conditions can be achieved (e.g. environment instantiation, sensors equipped by the robot). Thereby, comparisons between different approaches and configurations are facilitated. Furthermore, this solution reduces the subjects' recruitment costs without strongly hampering their natural expressiveness (due to the capacities offered by the simulator).

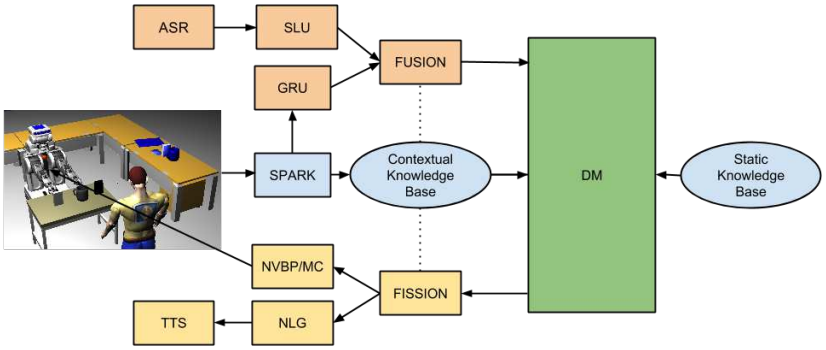

Fig. 4: Architecture of the multimodal and situated dialogue system.

4.2 Architecture

The multimodal dialogue architecture considered in our experiments is presented in Figure 4. Twelve components are responsible of the overall functioning of this dialogue system.

The four orange ones are those which are implicated in the user's input management, speech and gesture modalities in our case. Thus, the combination of the Google Web Speech API[3] for Automatic Speech Recognition (ASR) and a custom-defined grammar parser for Spoken Language Understanding (SLU) are used to perform speech recognition and understanding. The Gesture Recognition and Understanding (GRU) module simply catches the gesture-events generated by our spatial reasoner during the course of the interaction. Then, the Fusion module temporally aligns the monomodal inputs then merge them with custom-defined rules. Finally, the result of the fusion (i.e. N-best list of interpretation hypotheses and their related confidence scores) becomes the input of the multimodal DM.

The three blue components are responsible of the context modelling. SPARK, previously presented in 3, for both detecting the user gestures and generating the per-agent spatial facts (perspective taking) which are used to dynamically feed the contextual knowledge base. These two modules are responsible of per-agent knowledge modelling which allows the robot to reason over different perspectives on the world. Furthermore, we also make use of a static knowledge base containing the list of all available objects (even those not perceived) and their related static properties (e.g. color).

The four yellow components are dedicated to the output restitution. So, the Fission module splits the abstract system action into verbal and non-verbal ones. The spoken output is produced by chaining a template-based Natural Language Generation (NLG) module with a Text-To-Speech Synthesis (TTS) component based on the commercial Acapela TTS system[4]. The Non-verbal Behaviour Plan-

[3] https://www.google.com/intl/en/chrome/demos/speech.html
[4] http://www.acapela-group.com/index.html

ning and Motor Control (NVBP/MC) module produces arm gestures and head and body poses for the robot by translating the non-verbal action into a sequence of abstract actions, as defined in 2.3.

Finally, the green component is the DM, responsible for updating the internal belief state and to take the next robot decision. It is based on the POMDP-based Hidden Information State (HIS) framework [19] which has been adapted to the multimodal case here. In this setup, the belief state is represented by a set of partitions. Each partition represents a possible user command. The decision takes place into a more reduced summary space where RL algorithms are tractable. So, at each turn the system choose a summary action (e.g. inform, confirm, execute) and a heuristic-based method maps the summary action back to the master state (hand-crafted part).

Concerning the DM policy, the sample-efficient KTD-SARSA RL algorithm [25] was used in combination with the Bonus Greedy exploration scheme to enable the online learning of a dialogue policy from scratch. A reward function is defined to penalise the DM by -1 for each dialogue turn and reward it by $+20$ if the right command is performed at the end of the interaction, 0 otherwise. More details about this setup are available in [27,28].

4.3 Experimental Setup & Results

In this "proof of concept" study we chose to deal with a limited expert panel, composed of 6 subjects (2 females and 4 males of around 25 years old), in order to focus on the capacity of the system to learn from scratch using a limited set of interactions. The advantage is that the collected data sufficiently explore the state and action spaces during the online learning to be exploited in offline learning (using batch samples).

At the beginning of each dialogue, a specific goal (here a command) is randomly generated taking into account the simulated environment settings and the current interaction history in order to select a possible command. For example, "You want the robot to give you the white book on the kitchen table". No experimenter has any idea of the chosen configuration of the system with which he is interacting. So, we basically compare a hand-crafted expert dialogue policy (noted HDC) to a learned one (noted LEARNED). The latter was trained using a small set of expert users which first performed 60 dialogues in an online learning setting.

In the complete multimodal architecture, each interaction takes from 7 to 10 minutes to complete (objects detection, robot movements, etc.). So, without loss of generality, a practical workaround to speed-up the testing process consisted in using a fixed representation of the scene (a screenshot from the human point of view) and a web-based multimodal GUI instead of the full simulation setup. Overall, 84 dialogues for both the two proposed systems were recorded with 6 distinct subjects. At the end of each interaction users evaluated the system in terms of task completion. The learned policy were configured to act greedily according to the value function. Results are those gathered in test condition where exploration is not allowed. All the dialogues were recorded both in terms of audio

and various kinds of meta-information (e.g. ASR N-Best list, dialogue manager detected gestures and related timestamps, etc.) but also high level annotations (e.g. environment settings, pursued goal, task success). As an illustration, a short interaction, translated from French, is given in Table 1.

R1	DA	hello()
	NLG/TTS	Can I help you ?
U1	ASR	Can you put the book in my bedroom?
	SLU	inform(action=move,desc=in,room=bedroom)
R2	DA	confreq(type=book,position)
	NLG/TTS	Sorry but where is the book you are talking about?
U3	ASR	I am talking about this one
	SLU	inform(idobj=?)
	GRU	pointsAt BLUE_BOOK 1395848705.31
R3	DA	execute(action=move,destination=bedroom_bedsidetable, idobj=BLUE_BOOK,position=livingroom_table,type=book, color=blue)
	NVBP/MC	move(BLUE_BOOK,livingroom_table,bedroom_bedsidetable)
	NLG/TTS	Ok, I will put the blue book on your bedside table

Table 1: Example of a multimodal dialogue.

The results obtained are 14.3 for the HDC method and 17.6 for the LEARNED one. These results are given in terms of mean discounted cumulative rewards [18]. According to the reward function definition, this metric expresses in a single real value the two variables of improvement, namely the success rate (accuracy) and the number of turns until dialogue end (time efficiency). So, here the HDC policy manages the dialogue with 86% of success rate in an average of 4.8 turns against respectively 93% and 2.9 turns for the LEARNED one. The difference observed between the two methods can be mainly explained by a more accurate and less frequent usage of request of confirmation as well as an expected more fined-grained uncertainty management for the LEARNED method. Thus, these results clearly both demonstrates the ability of the overall architecture (simulation software + multimodal dialogue system) to learn an efficient dialogue policy using few dialogue examples and shows the interest of considering RL methods rather than a hand-crafted fixed and suboptimal policy. Indeed, only 60 training dialogues are enough to outperform the HDC by more than 3 points.

5 Summary and Future Work

In this paper we show how the MORSE simulator is used to build a scenario for HRI and how we used a robotic system along with this simulator to provide situated data to train the dialogue system. Using the MORSE simulator along with a robotic system was very helpful for us as it allows several partners to work with the same environment even being at different physical places and allows to train the system without using the actual robot, making it much easier

for trainers. We believe this configuration is close enough to reality to efficiently train the dialogue system. Anyhow as we have not yet deployed the dialogue system on the robotic platform this affirmation still needs to be proved. These metrics will be carried out in a future work.

Acknowledgments This work has been supported by l'Agence Nationale pour la Recherche under project reference ANR-12-CORD-0021 (MaRDi).

References

1. L. Lucignano, F. Cutugno, S. Rossi, and A. Finzi, "A dialogue system for multimodal human-robot interaction," in *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 197–204, ACM, 2013.
2. R. Stiefelhagen, H. K. Ekenel, C. Fugen, P. Gieselmann, H. Holzapfel, F. Kraft, K. Nickel, M. Voit, and A. Waibel, "Enabling multimodal human–robot interaction for the karlsruhe humanoid robot," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 840–851, 2007.
3. D. K. Byron and E. Fosler-Lussier, "The osu quake 2004 corpus of two-party situated problem-solving dialogs," in *Proceedings of the 15th Language Resources and Evaluation Conference (LREC'06)*, 2006.
4. T. Prommer, H. Holzapfel, and A. Waibel, "Rapid simulation-driven reinforcement learning of multimodal dialog strategies in human-robot interaction.," in *INTERSPEECH*, 2006.
5. V. Rieser and O. Lemon, "Learning effective multimodal dialogue strategies from wizard-of-oz data: Bootstrapping and evaluation.," in *ACL*, pp. 638–646, 2008.
6. R. B. Rusu, A. Maldonado, M. Beetz, and B. P. Gerkey, "Extending Player/Stage/Gazebo towards cognitive robots acting in ubiquitous sensor-equipped environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop for Network Robot Systems*, 2007.
7. S. Nakaoka, S. Hattori, F. KANEHIRO, S. Kajita, and H. Hirukawa, "Constraint-based dynamics simulator for humanoid robots with shock absorbing mechanisms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS2007)*, 2007.
8. R. Diankov, *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute, August 2010.
9. M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, "Virtual robot experimentation platform v-rep: a versatile 3d robot simulator," in *Proceedings of the Second international conference on Simulation, modeling, and programming for autonomous robots*, SIMPAR'10, (Berlin, Heidelberg), pp. 51–62, Springer-Verlag, 2010.
10. M. Lewis, J. Wang, and S. Hughes, "Usarsim : Simulation for the study of human-robot interaction," *Journal of Cognitive Engineering and Decision Making*, vol. 2007, pp. 98–120, 2007.
11. G. Echeverria, S. Lemaignan, A. Degroote, S. Lacroix, M. Karg, P. Koch, C. Lesire, and S. Stinckwich, "Simulating complex robotic scenarios with morse," in *SIMPAR*, pp. 197–208, 2012.
12. S. Lemaignan, M. Hanheide, M. Karg, H. Khambhaita, L. Kunze, F. Lier, I. Ltkebohle, and G. Milliez, "Simulation and hri: Recent perspectives," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, (SIMPAR2014)*, 2014.

13. G. Milliez, M. Warnier, A. Clodic, and R. Alami, "A framework for endowing interactive robot with reasoning capabilities about perspective-taking and belief management.," in *Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication*, 2014.
14. S. Alili, V. Montreuil, and R. Alami, "HATP Task Planner for social behavior control in Autonomous Robotic Systems for HRI," in *The 9th International Symposium on Distributed Autonomous Robotic Systems*, 2008.
15. D. Nau, T. C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman, "SHOP2: An HTN Planning System," *Journal of Artificial Intelligence Research*, pp. 379–404, 2003.
16. E. A. Sisbot, A. Clodic, R. Alami, and M. Ransan, "Supervision and Motion Planning for a Mobile Manipulator Interacting with Humans," 2008.
17. M. Fiore, A. Clodic, and R. Alami, "On planning and task achievement modalities for human-robot collaboration," in *International Symposium on Experimental Robotics, Marrakech/Essaouira, June 1518, 2014*, 2014.
18. R. Sutton and A. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998.
19. S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The hidden information state model: A practical framework for pomdp-based spoken dialogue management," *Computer Speech and Language*, vol. 24, no. 2, pp. 150–174, 2010.
20. B. Thomson and S. Young, "Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems," *Computer Speech and Language*, vol. 24, no. 4, pp. 562–588, 2010.
21. F. Pinault and F. Lefèvre, "Unsupervised clustering of probability distributions of semantic graphs for pomdp based spoken dialogue systems with summary space," in *IJCAI 7th KRPDS Workshop*, 2011.
22. N. Roy, J. Pineau, and S. Thrun, "Spoken dialogue management using probabilistic reasoning," in *ACL*, 2000.
23. M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu, and S. Young, "Gaussian processes for fast policy optimisation of pomdp-based dialogue managers," in *SIGDIAL*, 2010.
24. L. Sungjin and M. Eskenazi, "Incremental sparse bayesian method for online dialog strategy learning," *Journal on Selected Topics in Signal Processing*, vol. 6, pp. 903–916, 2012.
25. L. Daubigney, M. Geist, S. Chandramohan, and O. Pietquin, "A comprehensive reinforcement learning framework for dialogue management optimization," *Journal on Selected Topics in Signal Processing*, vol. 6, no. 8, pp. 891–902, 2012.
26. M. Geist and O. Pietquin, "Kalman temporal differences," *Journal of Artificial Intelligence Research (JAIR)*, vol. 39, pp. 483–532, Sept. 2010.
27. E. Ferreira and F. Lefèvre, "Social signal and user adaptation in reinforcement learning-based dialogue management," in *Proceedings of the 2nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication*, pp. 61–69, ACM, 2013.
28. E. Ferreira and F. Lefèvre, "Expert-based reward shaping and exploration scheme for boosting policy learning of dialogue management," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pp. 108–113, IEEE, 2013.