A Dialogue System for Multimodal Human-Robot Interaction

Lorenzo Lucignano, Francesco Cutugno, Silvia Rossi, Alberto Finzi
DIETI, Univ. di Napoli "Federico II"
Via Claudio 21, I-80125, Napoli, Italy
{lor.lucignano,cutugno,silvia.rossi,alberto.finzi}@unina.it

ABSTRACT

This paper presents a POMDP-based dialogue system for multimodal human-robot interaction (HRI). Our aim is to exploit a dialogical paradigm to allow a natural and robust interaction between the human and the robot. The proposed dialogue system should improve the robustness and the flexibility of the overall interactive system, including multimodal fusion, interpretation, and decision-making. The dialogue is represented as a Partially Observable Markov Decision Process (POMDPs) to cast the inherent communication ambiguity and noise into the dialogue model. POMDPs have been used in spoken dialogue systems, mainly for tourist information services, but their application to multimodal human-robot interaction is novel. This paper presents the proposed model for dialogue representation and the methodology used to compute a dialogue strategy. The whole architecture has been integrated on a mobile robot platform and has been tested in a human-robot interaction scenario to assess the overall performances with respect to baseline controllers.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Input devices and strategies, Interaction styles*; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Markov processes*

Keywords

POMDP-based Dialogue Management; Multimodal HRI

1. INTRODUCTION

Multimodal communication is a key factor in human-robot interaction making the robot companion able to adequately interpret and react to human actions [1]. Many works have shown how merging information, provided by different input channels, increases the performance of system and allows a natural communication experience [2, 3, 4, 5, 6, 7, 8]. This is also supported by studies in cognitive psychology [9]. In order to cohabit, collaborate and share the common workspace with humans, a robotic system needs the ability of recognizing human commands and activities, understanding user's intentions and deciding a rational interaction strategy, even in presence of noise and low level of confidence. In this paper, we propose to face these issues deploying a dialogical paradigm in the context of multimodal human-robot interaction. The main idea is that a natural and flexible interaction between the human and the robot can be represented and managed as a dialogue involving multiple modalities: speech, gestures, moviments, body postures, emotions etc.. The interactive, multimodal, dialogue flow should allow both the human and the robotic system to interpret and disambiguate contexts and intentions. Dialogue systems have been proposed in HRI, mainly focussing on the speech modality [10, 11, 12] or considering it as the dominant one [13, 14]. In contrast, we propose a dialogue system that exploits all the available modalities to contextualize, interpret, and orchestrate the overall interaction process.

For this purpose, we deploy a probabilistic approach. All the approaches to dialogue management, but the probabilistic one, assume the full observability of the dialogue state which is realized by means of the fusion results. These approaches commonly overlook the handling of the uncertainty [15]. As a result, complex error recovery procedures are needed to handle misunderstanding or failures, while the associated policies tend to make the dialogue repetitive and to increase the amount of human interventions. Furthermore, the context-free fusion of multiple modalities could fail when the inputs are contradictory, hence in this case the interpretation should depend on the state of dialogue.

Representing the multimodal dialogue as a Partially Observable Markov Decision Process (POMDP) potentially provides an effective way to cast the uncertainty into the interaction model and to select the machine actions according to the levels of confidence provided by the sensor fusion process. The key idea is that system cannot know the real dialogue state, hence it can keep a probability distribution over the possible states.

In literature, POMDPs have been successfully used for spoken dialogue system in HCI. For example, in [16], the authors presents a framework, called Hidden Information State, for handling uncertainty in spoken dialogue applications considering the case study of a tourist information system. As for HRI, in [12] the authors propose a prototype nursing home robot endowed with a POMDP-based spoken dialogue system.

ICMI'13 December 09-13 2013, Sydney, NSW, Australia
Copyright 2013 ACM 978-1-4503-2129-7/13/12 ...$15.00.

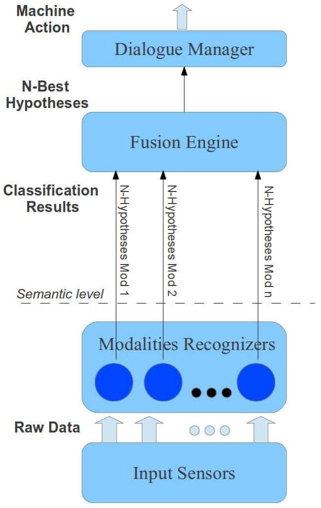

Figure 1: Architecture for Multimodal Communication

In this paper, we present a novel POMDP-based dialogue system for HRI that integrates multimodal communication, probabilistic dialogue management, and contextual information. The main problem of POMDPs is that computing an exact solution for large POMDP is unfeasible [17], therefore we have developed an approximated solution which is suitable for our domain. The dialogue system along with the overall multimodal interaction framework has been integrated on a mobile robot platform and tested in a human-robot interaction scenario. In this context, we have evaluated the dialogue system performance with respect to some baselines controllers in terms of classification rates, dialogue duration, and amount of interventions.

The paper is organized as follow: in Section 2 we introduce the system architecture; in Section 3 we illustrate the POMDP dialogue manager; in Section 4 we introduce a Pick-and-Place testing scenario providing the evaluation results; finally, in Section 5, we discuss conclusions and possible future developments

2. MULTIMODAL ARCHITECTURE

The dialogue system presented in this paper is the upper layer of a modular architecture for multimodal interaction (Figure 1). The proposed architecture is consistent with the abstract pattern proposed in [9].

The key feature of this architecture is that each layer provides the next one with a N-best list of possible interpretations, in order to solve in cascade the ambiguities at the upper layers of the system. The lower layer contains the classifiers of the single modalities. Currently, gesture and speech recognition are available. Speech recognition is performed by the Google Speech Api, by sending the registration of human's utterance to the server. The server returns a list of possible phrases with the related confidence rates. The semantic interpretations of each result are extracted by a SVM-based classifier, which returns a list of user's spoken actions in a structured representation. Gesture recognition is performed by analyzing data from a RGB-D camera. The set of features is made up of 17 elements, which includes the 3D coordinates of the body joints, the 3D angles between the joints, and the hand status (open, closed, pointing). The hand status is distinguished basing on the recognition of different color blobs on a glove. The classification process is modeled by Hidden Markov Models, which have been trained using a training set of 1280 elements. The result of the classification process, as for the speech case, is again a list of gestures, each with its own score. The results provided by each single recognizer is the input of a fusion engine. The fusion process is based on a late fusion statistical approach and provides a context-free integration of the multiple inputs [18]. The Fusion Engine *temporally* aligns the monomodal results, establishing a temporal window in which all the inputs are *parallel* and contribute to the fusion; then it performs the merging using a SVM classifier. Finally, the N-best list of fusion results becomes the input of the multimodal dialogue manager presented in this paper. It performs the coordination of the dialogue flow and accomplishes the semantic interpretation of the observations/actions according to the context and the inner knowledge. As previously said, when the single modalities are discordant, the fusion engine provides a low confident result, or in the worst case, it straight returns the monomodal scores. In this case, the dialogue manager has to choose the right action, while keeping every possible hypotheses about user's intentions. Moreover, the dialogue manager deals with the fusion of sequential inputs, which are not processed by the fusion engine since they do not belong to the same temporal window. From this point of view, the dialogue manager works as a long time scale fusion module. More details about the classifiers employed in this work can be found in [19].

3. POMDP DIALOGUE MANAGER

Dialogue management is mainly conceived to deal with some issues derived from the communication task. Due to the nature of the task and the presence of noise on each channel, it is possible to identify several sources of *ambiguity*, like misunderstanding of human actions or commands, multiple interpretations of a particular observation or non-deterministic effects of a machine action. For these reasons, our approach is to use a POMDP formalization of a dialogue process. A dialogue is made up by multiple flows, which are the possible branches of the conversation. A dialogue flow contains the nodes, which represent the situations that may occur, and defines the turns of the dialogue. Each node is characterized by the observable user's actions. Since the machine does not know the user's intention, at each time step it could be in multiple situations, hence it should consider multiple hypotheses about the current dialogue state. Hence, the choice of the next machine action needs to take into account this uncertainty.

The POMDP model of our system is a tuple $(S, A_m, T(\cdot,\cdot,\cdot), r(\cdot,\cdot), O, Z(\cdot,\cdot), b_0)$. $S = S_{flow} \times S_{node} \times A_u$ is the set of states. A state is a triple $s =< s_{flow}, s_{node}, a_u >$ where s_{flow} is the ID of a dialogue flow, s_{node} the ID of a node in the dialogue flow and a_u is the last user action, such as a monomodal act or a multimodal one. A_m is the set of machine actions. The set contains the execution actions, which are the interpretations of the user's commands or activities, and the control actions, which are useful to get confirmations or decisions by the user. Since the effect of machine actions are non-deterministic, $T(s', s, a_m)$ is the transition probabil-

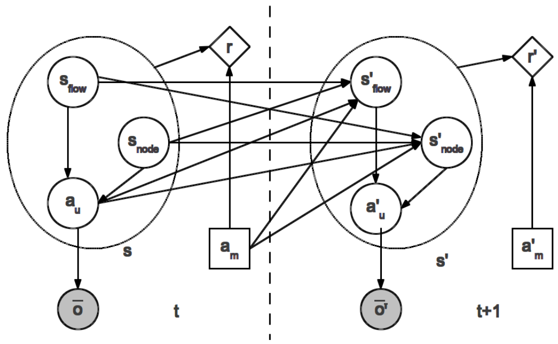

Figure 2: The POMDP dialogue model represented as a Bayesian Network. The white circles are the hidden variables, while gray circles denote the observations. The machine action a_m depends on the current belief b.

ity $P(s'|s, a_m)$ and R is the reward function $R(s, a_m) \in \mathbb{R}$. O is the set of the observations, which are N-best lists of hypotheses about an user's action $\bar{o} = [< a_u^1, p_1 > \dots < a_u^n, p_n >]$, where $p_i = P(a_u^i|\bar{o})$, $a_u^i \in A_u$. The observations are provided by fusion layer. $Z(s, \bar{o})$ is the observation probability $P(\bar{o}|s)$. Finally, b_0 is the initial belief state.

The probability distribution among the states is called belief state b, and $b(s)$ is the probability of being in state s. The system works as a classic POMDP: at each time step, the system has a belief over states and executes an action according to a policy. Then it goes in another hidden state, and gets an observation. Finally, it updates the belief. The factorization of the state allows some independence assumptions, which simplify the conditional dependencies that govern the belief update function. The transition probability is the following:

$$T(s', s, a_m) = P(s'\bar{s}, a_m) \approx P(a_u^{'}\bar{s}_{flow}^{'}, s_{node}^{'}) \cdot$$
$$P(s_{flow}^{'}, s_{node}^{'}\bar{s}_{flow}, s_{node}, a_u, a_m), \quad (1)$$

that is, the next user's action a'_u depends only on the next dialogue s'_{flow} and the next nodes s'_{node}, which are determined by the current state $< s_{flow}, s_{node}, a_u >$ and the last machine action a_m. The observation probability is assumed to be $Z(s, \bar{o}) = P(\bar{o}'|s', a_m) \approx P(\bar{o}'|a_u^{'})$, hence the next observation depends on the next user's action. The Figure 2 shows the dependency graph. The above functions implies the formulation of the belief update equation:

$$b(s_{flow}^{t+1}, s_{node}^{t+1}, a_u^{t+1}) = k \cdot P(\bar{o}^{t+1}\bar{a}_u^{t+1}) \cdot P(a_u^{t+1}\bar{s}_{flow}^{t+1}, s_{node}^{t+1}) \cdot$$
$$\sum_{(s_{flow}^t, s_{node}^t, a_u^t) \in b} (P(s_{flow}^{t+1}, s_{node}^{t+1}\bar{s}_{flow}^t, s_{node}^t, a_u^t, a_m^t) \cdot$$
$$b(s_{flow}^t, s_{node}^t, a_u^t)) \quad (2)$$

According to [20], by assuming that the prior probability $P(a_u)$ is constant for each user action, the observation model could be derived for the results of the fusion engine $P(\bar{o}|a_u) = P(\bar{o}|a_u = a_u^i) = k_0 \cdot P(a_u^i|\bar{o})$. In this way, the constant k_0 can be absorbed by the constant k in the belief update function. By this, the quality of the classifier plays an important role, and it will become a variable of the evaluation tests. Furthermore, it is important to notice that the multiplication of the observation model and the user action model results in a re-estimation of the scores of N-best list according to the current belief.

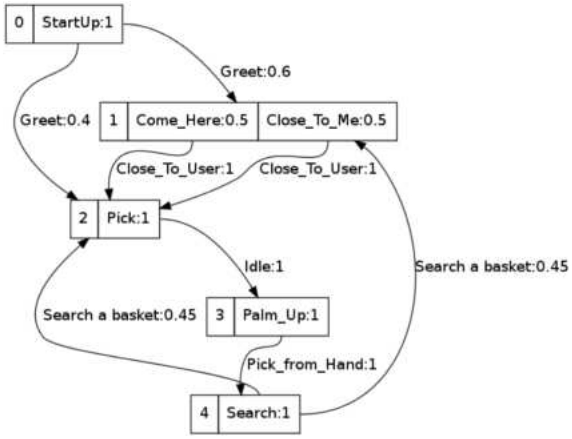

Figure 3: Dialogue Flow Example. The node 1 is charactered by two possible observations "Come_Here" and "Close_to_Me". The related machine action is to go near the user (Close_To_User). This machine action triggers a transition to the node 2, in which machine expects that user asks to pick something.

The dialogue models are provided by the developer as graph-based specifications written in XML. An example of a dialogue graph describing a simple interaction scenario is shown in Figure 3. These dialogue models can be incrementally added to the system. The XML syntax allows us to link together the flows to obtain complex dialogue models.

3.1 Policy Search

Once the model is defined, the next step is to find a policy π, a function which associates a machine action to each belief state. Usually, a dialogue strategy is evaluated in term of number of turns required to complete user's requests, number of failures or control actions etc. [21]. We need to find a trade-off between reliability in action execution and usability: a machine that continously asks for agreement is frustrating for the user, while a machine that tries to guess the user intentions without any feedback from the user can be dangerous. Since the probabilistic model is already known, the policy is not learned, but foreordained by an iterative algorithm which maximize the expected discounted value function over an infinite horizon T [22], i.e. $V_T^{\pi^*}(b) = \gamma \cdot \max_{a \in A}[\sum_{s \in S} b(s) \cdot r(s, a) + \int V_{T-1}^{\pi^*}(b') \cdot P(b'|b, a)\, db']$, with $V_0^{\pi^*}(b) = \gamma \cdot \max_{a \in A}[\sum_{s \in S} b(s) \cdot r(s, a)]$. Although computing a optimal policy is possible for small POMDPs under the assumpion that the state space, the action space, the space of observations and the planning horizon are finite, computing an exact optimal solution is notoriously intractable when dealing with a real-world POMDP.

In spoken dialogue systems, the use of belief state compression for approximating POMDP policy is a common solution. These are characterized by the structure of the summarized space, for example deploying partitions and ontology trees

[23], or using measurements of entropy [12]. Some remarks on the nature of dialogue task led us to the design of an algorithm for an approximated solution which works on a summary representation of a POMDP. This method is based on both Augmented MDP and Point Based Value Iteration [22, 17]. The first observation is that when the robot has multiple hypotheses and has to select an action, it could perform either the most probable action or a control action. By this, for the policy optimization, the action set can contain only control actions along with one action called *Do_act*, which summarizes the execution of the most probable action. The choice of a particular action will depend on some thresholds of the uncertainty level, that will be discovered by algorithm. The identification of such thresholds leads to another issue. For the purpose of action selection, more than the dialogue state, it matters the amount of confidence for each actual hypotheses and for the next ones. For this reason, the belief state could be summarized in a lower dimensional vector, containing only a partial estimate of the information quantity.

The policy optimization is therefore performed exploiting a Summary Belief MDP $\left(\tilde{B}, \tilde{A}, \tilde{T}^b, \tilde{R}^b, \tilde{b}_0\right)$. Here, $\tilde{B}$ is the set of states. A state $\tilde{b}$ is a triple generated from a belief state b that contains the probability of the first hypotheses, the probability of the second one, and a measure of compatibility among the actions of all hypotheses. The third component maps how many hypotheses share the action linked to the top hypothesis to identify those situations in which multiple dialogue states support the same machine action. $\tilde{A}$ is the set of machine actions and it contains all the control actions, for example request or confirmations, and the "*Do_act*" action, whose meaning is to perform the execution action related to the top hypothesis. Typically, the control actions are far less than execution actions, hence this choice significantly reduces the action space. $\tilde{T}^b$ is the belief transition and $\tilde{R}^b$ is the reward function. Since the function $\tilde{T}^b$ and $\tilde{R}^b$ are unknown, they need to be constructed by a frequency statistic, as for the Augmented MDP algorithm. Finally, $\tilde{b}_0 \in \tilde{B}$ represents the initial state.

As we said above, the optimization algorithm is based on a Point Based Value Iteration and Augmented MDP. The first step of this is to build a set of points $\tilde{b}_1 \dots \tilde{b}_n \in \mathfrak{B} \subseteq \tilde{B}$ on which the policy will be optimized. The dimension of the set and the smallest distance between points are key input variables. The aim is to get a set *representative* enough of the whole state space, where the small size or close clustering can speed up the optimization process, but this can yield to a poor quality policy. By contrast, a large set can cause the opposite effects. Once the point-selection phase is finished, it is necessary to learn, for each point, the function $\tilde{T}_i^b$ and $\tilde{R}_i{}^b$ through a sampling phase in which the dynamics of summary space are estimated. Finally, the optimization algorithm is performed to get a policy. The optimization is done only for the points in the basket, bounding the complexity of the algorithm. In the generated policy we have the list of the summarized belief point $\tilde{b}_1 \dots \tilde{b}_n$ and the relative machine actions to perform. At runtime, the choice of the actions is made by summarizing the current belief state and searching the closest one in the list. The idea is that if an action is good for a point, it might be good enough for the points which are close.

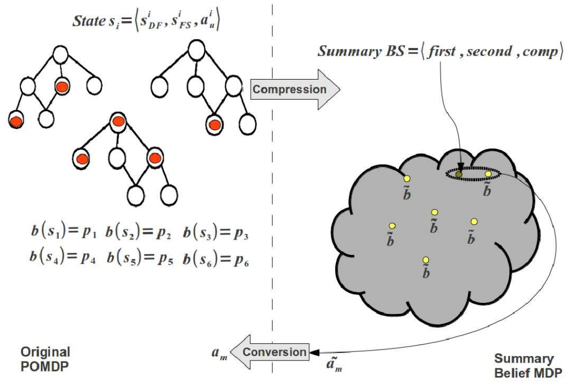
State $s_i = \langle s^i_{DF}, s^i_{FS}, a^i_u \rangle$
Compression
Summary BS = ⟨first, second, comp⟩
$b(s_1)=p_1$ $b(s_2)=p_2$ $b(s_3)=p_3$
$b(s_4)=p_4$ $b(s_5)=p_5$ $b(s_6)=p_6$
$\tilde{b}$
Original POMDP
a_m
Conversion
$\tilde{a}_m$
Summary Belief MDP

Figure 4: Runtime action selection

4. CASE STUDY

To assess our system, we designed a test-bed scenario which is an instance of a common Pick-Place-Carry situation. The robot and the user are in a closed room, in which there are red or yellow balls and numbered baskets. The user can interact with the robot using gestures or body movements, while the robot has a list of user dialogue models describing possible flows of commands or movements. Each gesture can be associated with one or more meanings, hence ambiguities are possible. The meaning can be made clear, according to the dialogue context, using the dialogue models provided to the system. However, some user's acts are not explicit commands, therefore the system should understand the human's intention and should support the human activity by its planning and control skills.

4.1 Off ine test

The aim of these tests is to assess the performance of computed policy compared with a greedy one and two handcrafted ones. Our working hypothesis is that dialogue can improve the quality of the communication providing a tradeoff between the amount of requests and the correctness of execution. For example, in some situations, the execution of an action could increase the belief about a single hypothesis without annoying the user. These kinds of behaviors are usually complex to achieve using local strategies while policy optimization can be a suitable method to generate them.

Test setup. The system is provided with 5 dialogue flows, for a total of 68 states. The user's actions are 12 and the possible machine action are 16 plus 2 control actions, which are *Request*, for asking a confirmation on the current most probable action, and *ChooseAmong2*, for deciding between the two most probable actions. The main feature of those dialogue models is that many ambiguities may arise even in the presence of high recognition rates. The test sessions are composed of 100 interactions. The reward function $r(s, a_m)$ provides a positive reward (+10) for taking the correct action or a large penalty (-20) otherwise. The control actions result in little penalties. The reward for the *ChooseAmong2* action results in a huge negative value to avoid the selection of the action when there is only one hypothesis or when the two most probable actions coincide.

During the tests, the misunderstanding rate of the classification result is modulated by the parameter λ. The confidence c of selecting the user actions is drawn from an expo-

Machine action	Condition	Reward
execute(a_m)	$act(s) = a_m$	+10
execute(a_m)	$act(s) \neq a_m$	-20
request(a_m)	$act(s) = a_m$	-0.5
request(a_m)	$act(s) \neq a_m$	-3
chooseAmong2 (a_m^1,a_m^2)	$a_m^1 \neq Null \wedge a_m^2 \neq$ $Null \wedge a_m^1 \neq a_m^2$	-1
chooseAmong2 (a_m^1,a_m^2)	$a_m^1 = Null \vee a_m^2 =$ $Null \vee a_m^1 = a_m^2$	$-\infty$

Table 1: Reward Function

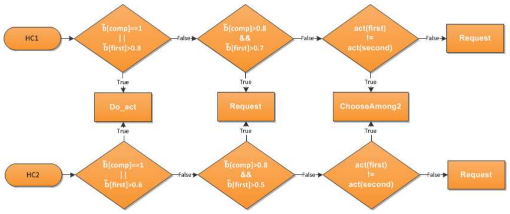
HC1
b̃[comp]==1 \|\| b̃[first]>0.8
False
b̃[comp]>0.8 && b̃[first]>0.7
False
act(first) != act(second)
False
Request
True
True
True
Do_act
Request
ChooseAmong2
True
True
True
HC2
b̃[comp]==1 \|\| b̃[first]>0.6
False
b̃[comp]>0.8 && b̃[first]>0.5
False
act(first) != act(second)
False
Request

Figure 5: Handcrafted conservative (up) and brave policies (down).

nential distribution[24]. The slope of such function is determined by the parameter $\lambda > 0$, $p_\lambda(c) = \frac{\lambda e^{\lambda c}}{e^{\lambda}-1}$. The size of $\mathfrak{B}$ is 200 summary points, while the planning horizon and the discount factor are respectively $T = 100$ and $\gamma = 0.9$. In this way we give almost the same weight to all the future actions in the horizon.

Test results. To assess the quality of the automated planning, we compared the performance of a greedy policy, two handcrafted policies, and the optimized policy. The greedy policy selects the action with the immediate highest expected reward, i.e. $V(b) = \max_{a \in A}[\sum_{s \in S} b(s) \cdot r(s,a)]$. The first handcrafted policy is conservative, whereas the second one is more brave (see Figure 5). The conservative policy selects the most probable action choice if the probability is greater than 0.8, otherwise, if the probability is between 0.7 and 0.8 and it is supported (compatibility among the actions of all the hypotheses greater than 0.8), it asks for a confirmation; in the other cases it asks the user to select between the best two actions (when at least two actions are available). Instead, the brave policy selects the most probable action with probability greater than 0.6, asks for a confirmation when the probability is between 0.5 and 0.6 with compatibility greater than 0.8; otherwise it asks the user to choose between the two best action candidates (when available).

The most evident result is that the average reward for the four policies are almost the same, although the chart exhibits a loss for POMDP-DM in the test with a high recognition rate (Figure 6). In this situation, the policy optimization causes an over-fitting, which results in bad exploration of all belief space during the point selection phase. Nevertheless, by analyzing the actions performed, it comes out that the POMDP-DM solution achieves this performance keeping the number of control actions smaller than the other solutions. Table 2 reports the average percentages of additional time achieved by the policies with respect to the whole dialogue duration. The first and the second columns contain, respectively, the average percentages of user actions due to requests and due to bad interpretations. The values of the third column are obtained as a sum of the former ones. The greedy policy and the handcrafted ones handle the uncertainty by asking for the user's confirmation repeatedly in so extending the duration of the dialogue. This behavior is especially visible in low recognition rate situations. The POMDP-DM solution can benefit of the horizon and a statistical knowledge about the effects of actions and the future states. This makes the system more "courageous", even if it could lead to possible rejection from the user. However, the rejection can be interpreted as an implicit request, since it permits to resolve uncertainty in a shorter manner. The average rates of error is 5,25% , which is adequate for a system with some degrees of autonomy, while, in contrast, it seems that spending about 20% of dialogue in confirmations and requests is too much (Table 2).

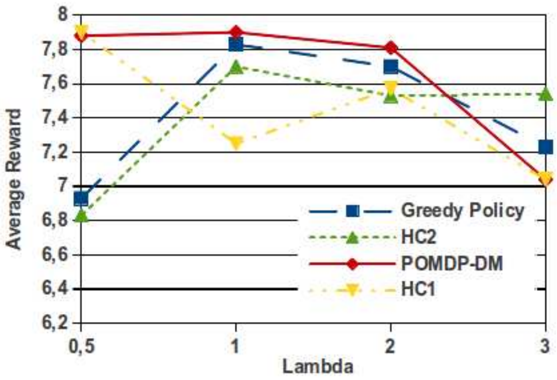

Figure 6: Average reward for POMDP-DM and baseline policies.

	Requests	Failures	Total Amount
Greedy	18,5	3,5	22
POMDP-DM	9,25	5,25	14,5
HC1	26,25	1,25	27,5
HC2	17	4,25	21,25

Table 2: Average percentage of additional time.

4.2 Online Test

The online test highlights the contribution of the context-aware dialogue management in the communication task. Mainly, dialogue models and N-best lists should allow the system to recover correct user's actions even if their ranks are low. The contextual information should improve the recognition performance and consequently reduce the duration of the dialogue when compared with the one-best solution. The trails have been conducted with 10 real users in order to collect a preliminary evaluation of the system and to record some suggestions useful for future enhancement.

Test setup. For the online test we have asked 10 people, 7 male and 3 female, to interact with the system. People were provided with a Bluetooth microphone and with a colored glove for hand status recognition. The task was to collect 10 balls and to place them somewhere.

The available gestures (depicted in Figure 7), the 9 possible speech commands, and the 2 composed patterns (”Pick” + Pointing, ”Place” + Pointing) has been shown to the testers, however the users had no information about preloaded

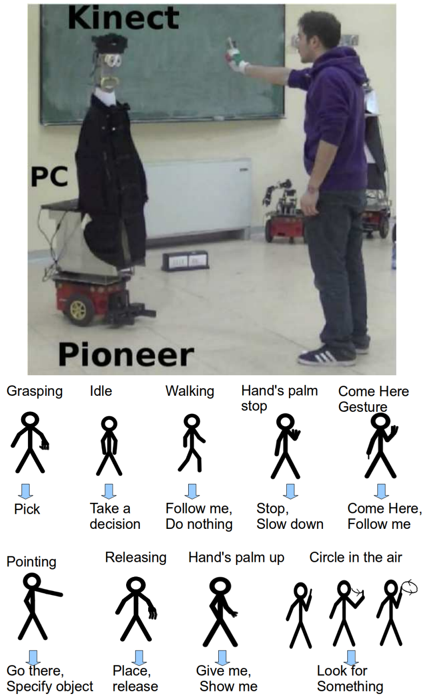

Figure 7: Robotic platform and available gestures with possible meanings

dialogue flows, which were 4 for a total of 40 states. The policy has been computed using the same setting of the offline tests. Since the gesture recognition rates are lower than the speech ones, we have preferred to perform the test three times: in the first place using only gestures, except for rejection performed through an utterance; the second time using only speech; the third time using both these modalities.

Test results. Both Tables 3 and Table 4 report the results of the trials. Those results aim to evaluate our approach from two points of view: on the one hand we want to analyse the contribution of the contextual information provided by the dialogue flows in the command recognition task; on the other hand, the aim is to estimate the quality of the dialogue when the interaction modality changes.

The first columns contain the correct classification rates of the user actions without the dialogue manager. A user action is well-classified if it takes the highest placement in the list provied by the fusion engine. The second columns report the correct classification rates of user actions using dialogue manager. As previously said, a user action can be interpreted in multiple ways, hence the dialogue manager has to assign the right semantic or it has to re-score the N-best list. The third columns contain the amount of control actions executed by the machine, while in the fourth columns the amount of the user actions performed during the dialogue are showed. In particular, the last data are obtained as the sum of the ”regular” actions and the ones due to bad interpretations. Finally, the last column reports the total amount of user actions. The gesture-based interaction achieves worse results than speech and multimodal interaction (Table 3). Due to changes of the light conditions and the distance of the user from the robot, the classification rates are affected by high inaccuracy. However, the dialogue manager increases the recognitions rate of approximately 7% by using the N-best hypotheses provided by the fusion engine. When the fusion engine does not correctly rate an action with the best score, usually that action takes a lower placement. In this situation, the dialogue model has the effect of rescaling the scores. In addition to this, the amount of user's interventions is high since the confidence on the user's actions is low and additional control actions are performed to enhance the confidence about multiple user's actions.

The test of the speech modality gets higher classification rates compared with a gesture-based interaction. The average rate without dialogue manager is 82,84%, which rises up to 85,47% using the dialogue manager. These results mainly arises from the high performance of the Google ASR service. The rare errors, caused by environmental noise, are well compensated by the dialogue manager, which in this case contextualizes utterances to the dialogue belief state rather than correcting the classification scores. Nevertheless, the speech communication lacks spatial information and this increases the number of turns to complete a single user's command. On average, the number of utterances for each trial is 28, which increases the average amount of total turns up to 33 when considering also the control actions. The multimodal interaction performs better than all the other cases (Tables 4). The classification rates are higher than the previous ones, and this supports the thesis that multiple inputs complement each other, improving the overall performance of the system, rather than generating conflicts and misunderstanding. In addition, the average duration of the dialogues is lower compared with single-modality trials, because the dialogue manager requests mainly help to disambiguate the meaning of a single action in relation with the current belief, hence the amount of interventions decreases remarkably.

The average classification rate is comparable with the results shown in recent the works of Burger [2] and Wu [25], which respectively reach 92% and 95% of correct interpretation scores for the multimodal communication. Finally, we asked users to fill a questionnaire, in order to collect impressions and advices for future development. The answer is a score from one to five, or rather *very bad*=1, *bad*=2, *sufficent*=3, *good*=4 and *very good*=5. The questions about gesture naturalness, speech naturalness, and multimodal naturalness aim to discover how much comfortable people feel to communicate by gestures, voice or both of them. The speech interaction and multimodal interaction achieve, respectively, good and very good score, since they are more similar to daily communication. The gestural commands seem artificial, however the work on gesture recognition should be considered as a first step towards human activity recognition. From this point of view, in the next development the machine should infer the action to perform by observing what the user is doing, moving from an active communication to an indirect one. The questions about efficiency are to evaluate how meaningful people consider their acts and how much confidence they give to the robot interpretation. The average scores agree with the classification rates of the tests, and for this reason the multimodal interaction gets the highest score.

5. CONCLUSIONS

In this paper we presented a dialogue-based approach to multimodal human-robot interaction (HRI). The aim was to exploit the dialogue paradigm to enable a natural, flexible,

	Classification rate without DM	Classification rate with DM	Requests	Gestures	Turns
User 1	58,82%	64,70%	5	34	39
User 2	64,51%	70,96%	9	31	40
User 3	50%	61,76%	8	34	42
User 4	33,33%	45%	9	60	69
User 5	46%	56%	9	50	59
User 6	60%	62,85%	8	35	43
User 7	58,82%	61,76%	6	34	40
User 8	65,78%	71,05%	9	38	47
User 9	65,62%	68,75%	7	32	39
User 10	51,11%	60%	7	45	52
Average	55,40%	62,28%	7,7	39,3	47

Table 3: Results of only gesture test

	Classification rate without DM	Classification rate with DM	Requests	Gestures+ Utterances	Turns
User 1	100%	100%	1	23	24
User 2	95,45%	95,45%	1	22	23
User 3	95,83%	95,83%	3	24	27
User 4	70,37%	74,07%	5	27	32
User 5	88,88%	88,88%	3	27	30
User 6	87,5%	87,5%	4	24	28
User 7	92%	92%	1	25	26
User 8	92,85%	92,85%	4	28	32
User 9	100%	100%	2	23	25
User 10	91,66%	91,66%	1	24	25
Average	91,45%	91,82%	2,5	24,7	27,2

Table 4: Results of multimodal test

Question	Average Score
Gesture naturalness	3,8
Gesture efficiency	3,4
Speech naturalness	4
Speech efficiency	4,2
Multimodal naturalness	4,8
Multimodal efficiency	4,4
Quality of dialogue (number of turn, requests, failures)	4,5

Table 5: Result of the questionnaires

and robust interaction and communication between the human and the robot companion. The novelty of our approach consists of using the dialogue manager to shape and to finalize the multimodal interpretation as well as to provide a strategic control of the dialogical behaviour. The proposed framework shows how multiple equally-weighted modalities, probabilistic dialogue management, and context can be combined to provide results in good performance and usability of the robotic agent. The POMDP model can handle such uncertainty by using prior knowledge provided by dialogue description and by tracking multiple hypotheses about the dialogue state. Furthermore, it simplifies error recovery without the need of designing complex mechanisms to handle misunderstanding. Since computing an exact solution for POMDP is intractable for problems with many states, we worked on finding a suitable representation of the dialogue as a POMDP and on designing an algorithm to solve it. The system evaluation illustrated that for our test cases the policy computed with the summary POMDP achieves better results comparing with baseline policies. In particular, it improves the quality of communication in terms of duration, number of requests and confirmations, whereas "local" policies tend to make the dialogue repetitive and to increase the amount of human interventions. The probabilistic approach also enhances the classification rate of the user's actions, since the hypotheses provided by the fusion engine are reevaluated according to the current belief and the dialogue models. The approval rating from the users is good and confirms that multimodal interaction is more comfortable and immediate than single-modality interaction, especially to complete the partial meanings of the single modalities. The tests and the questionnaires have highlighted some issues useful for future improvements. The bimodal interaction has achieved good results, since speech and gesture easily complement each other. However, it should be analyzed whether adding further communication modalities decreases performances or if the system can easily manage the increasing ambiguity. As far as the dialogue design is concerned, currently the probabilities of the dialogue models are provided by the dialogue designer, hence they could be incorrect. A better solution is to learn the probabilities from a corpus of dialogues, while leaving the models only for high level description, maybe providing a GUI toolkit along the lines of [26], or even to cast this uncertainty about dialogue models in the POMDP. Bayes-Adaptive POMDP models seem a suitable framework for this purpose, since they assume that transitions and observation probabilities are unknown or partially known [27]. Furthermore, the current policy optimization is performed offline and does not change during the dialogue. Hence, it could be useful to investigate the chance of enhancing the policy during the execution and according to the user's responses.

6. ACKNOWLEDGEMENTS

This work has been supported by EU FP7 SAPHARI under grant agreement no. ICT-287513, by the EU FP7 SHERPA under grant agreement no. ICT-600958, and by MIUR and EU under the PON OR.C.HE.S.T.R.A. project.

7. REFERENCES

[1] S. Oviatt. Advances in robust multimodal interface design. *Computer Graphics and Applications, IEEE*, 23(5):62 – 68, 2003.

[2] B Burger, I Ferrané, F Lerasle, and G Infantes. Two-handed gesture recognition and fusion with speech to command a robot. *Autonomous Robots*, pages 1–19, 2012.

[3] Rudiger Dillmann, Regine Becher, and Peter Steinhaus. Armar ii a learning and cooperative multimodal humanoid robot system. *Intern. Journal of Humanoid Robotics*, 01(01):143–155, 2004.

[4] M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. Match: An architecture for multimodal dialogue systems. In *Proc. of the Annual Meeting on Association for Computational Linguistics*, pages 376–383, 2002.

[5] Patrick McGuire, Jannik Fritsch, JJ Steil, F Rothling, GA Fink, S Wachsmuth, G Sagerer, and H Ritter. Multi-modal human-machine communication for instructing robot grasping tasks. In *IROS 2002*, volume 2, pages 1082–1088. IEEE, 2002.

[6] Heriberto Cuayahuitl and Ivana Kruijff-Korbayova. Towards learning human-robot dialogue policies combining speech and visual beliefs. In *Proc. of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop*, pages 133–140. 2011.

[7] Ivan Meza, Elia Perez, Lisset Salinas, Hector Aviles, and Luis A. Pineda. A multimodal dialogue system for playing the game guess the card. *Procesamiento de Lenguaje Natural*, 44, 2010.

[8] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *IROS 2004*, volume 3, pages 2422–2427. IEEE, 2004.

[9] Bruno Dumas, Denis Lalanne, and Sharon Oviatt. Multimodal interfaces: A survey of principles, models and frameworks. In *Human Machine Interaction*, volume 5440 of *Lecture Notes in Computer Science*, pages 3–26. Springer Berlin Heidelberg, 2009.

[10] Karolina Eliasson. Case-based techniques used for dialogue understanding and planning in a human-robot dialogue system. In *IJCAI 2007*, pages 1600–1605, 2007.

[11] L Seabra Lopes and A Teixeira. Human-robot interaction through spoken language dialogue. In *IROS 2000*, volume 1, pages 528–534. IEEE, 2000.

[12] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 93–100, 2000.

[13] Shuyin Li, Britta Wrede, and Gerhard Sagerer. A computational model of multi-modal grounding for human robot interaction. In *Proc. of SigDIAL '06*, pages 153–160, 2006.

[14] R. Stiefelhagen, H.K. Ekenel, C. Fugen, P. Gieselmann, H. Holzapfel, F. Kraft, K. Nickel, M. Voit, and Alex Waibel. Enabling multimodal human-robot interaction for the karlsruhe humanoid robot. *Robotics, IEEE Transactions on*, 23(5):840–851, 2007.

[15] T.H. Bui. Multimodal dialogue management-state of the art. Technical report, Centre for Telematics and Information Technology University of Twente, 2006.

[16] Steve Young, Milica Gasic, Simon Keizer, Francois Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech and Language*, 24(2):150 – 174, 2010.

[17] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI 2003*, Acapulco, Mexico, 2003. IJCAI.

[18] PradeepK. Atrey, M.Anwar Hossain, Abdulmotaleb El Saddik, and MohanS. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16:345–379, 2010.

[19] S. Rossi, E. Leone, M. Fiore, A. Finzi, and F. Cutugno. An extensible architecture for robust multimodal human-robot communication. In *IROS 2013*, page to appear. IEEE, 2013.

[20] Jason D Williams. A case study of applying decision theory in the real world: Pomdps and spoken dialog systems. In *Decision theory models for applications in artificial intelligence*. 2010.

[21] E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *Speech and Audio Processing, IEEE Transactions on*, 8(1):11 –23, jan 2000.

[22] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[23] Jason D Williams. *Partially observable Markov decision processes for spoken dialogue management*. PhD thesis, Cambridge University, 2006.

[24] O. Pietquin. *A framework for unsupervised learning of dialogue strategies*. Presses univ. de Louvain, 2004.

[25] Lizhong Wu, Sharon L Oviatt, and Philip R Cohen. From members to teams to committee-a robust approach to gestural and multimodal recognition. *Neural Networks, IEEE Transactions on*, 13(4):972–982, 2002.

[26] Bruno Dumas, Denis Lalanne, and Rolf Ingold. Hephaistk: a toolkit for rapid prototyping of multimodal interfaces. In *Proc. of ICMI 2009*, ICMI-MLMI '09, pages 231–232. ACM, 2009.

[27] Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-adaptive pomdps. In *Advances in Neural Information Processing Systems 20*, pages 1225–1232, 2007.