Optimizing Situated Dialogue Management in Unknown Environments

Heriberto Cuayáhuitl[1], *Nina Dethlefs*[2]

[1]German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
[2]University of Bremen, Germany
heriberto.cuayahuitl@dfki.de, dethlefs@uni-bremen.de

Abstract

We present a conversational learning agent that helps users navigate through complex and challenging spatial environments. The agent exhibits adaptive behaviour by learning spatially-aware dialogue actions while the user carries out the navigation task. To this end, we use Hierarchical Reinforcement Learning with relational representations to efficiently optimize dialogue actions tightly-coupled with spatial ones, and Bayesian networks to model the user's beliefs of the navigation environment. Since these beliefs are continuously changing, we induce the agent's behaviour in real time. Experimental results, using simulation, are encouraging by showing efficient adaptation to the user's navigation knowledge, specifically to the generated route and the intermediate locations to negotiate with the user.
Index Terms: spoken dialogue systems, situated interaction, reinforcement learning, hierarchical control, Bayesian networks

1. Introduction

Adaptive conversational interfaces provide a human user with information that is specifically suited to his or her own needs by taking the situational environment and individual properties of the user into account. Such interfaces need to perceive, reason about and act in their environment of operation to adapt sensibly to the user and conversational setting. This is especially true for situated agents which need to keep track of (a) features of the spatial setting, (b) the user's beliefs about this setting, and (c) dynamic changes of both occurring during the course of the interaction. We suggest to use Hierarchical Reinforcement Learning (HRL) to infer the agent's behaviour (or dialogue policy) from interactions with the environment. Reinforcement Learning (RL) is an attractive framework for optimizing sequential decision making, where situations are mapped to actions by maximizing a longterm reward signal [1, 2, 3]. HRL has the additional advantage of scaling up to large and complex problems [4]. A key requirement of the in-situ interaction scenario that we are addressing is that the user receives instructions as he/she carries out the task, so that adaptation to *unknown and continuously changing user prior knowledge* needs to occur 'on the fly'. This has a number of implications. Since not all properties of entities in the environment are known in advance, due to new world objects, navigational constraints, or the user's changing spatial knowledge, it is not possible to apply off-line learning. Instead, learning is required to occur in real time, after each user query. We tackle this problem by learning policies continuously often from a simulated environment, constantly updated with observations from the real environment.

Here we present a conversational agent for interactive route guidance based on constantly changing user prior knowledge to which the agent needs to adapt. To allow fast and flexible adaptation to the dynamically changing environment, we learn policies in real time, after each user query. This is achieved by applying relational representations to the agent's state-action space to effectively constrain it. For the agent to adapt to newly gained knowledge of the user online, we use Bayesian Networks (BNs) to track the user's beliefs about the environment during the navigation tasks. Finally, we follow and extend previous work [5] that has argued for a joint optimization of dialogue management and route planning into 'spatially-aware dialogue management'. In this way, the dialogue manager can draw on spatial knowledge to choose optimal dialogue actions, and the route planner can find optimal route instructions, given an unknown (or partially-known) and dynamic spatial environment. These aspects represent a substantial extension of previous work. In the rest of the paper we show that the combination of HRL with relational representations and Bayesian-based user beliefs is attractive for optimizing situated conversational behaviours into a unified, efficient and scalable framework.

2. Adaptive Situated Dialogue Management

The behaviour of situated dialogue managers is strongly influenced by the domain in which they operate, in our case the wayfinding domain. For instance, there can be multiple routes to guide the user from an origin to a destination, e.g. the easiest/shortest route to follow, the route simplest to describe, etc., and it is not trivial to decide which route is the best, given the current user and spatial environment. Such routes can be provided at different degrees of granularity, the best level will depend on the users' knowledge of the navigation environment and the complexity of the route. In other words, in order to supply optimal route instructions for individual users, the system not only needs information about the user's prior knowledge of the environment, but also about the space in which the user navigates. In particular, the dialogue manager must deal with questions such as *When to present information? When to ask for users' prior knowledge? What information to present according to the dialogue history and spatial environment?* Regarding the first two questions, the system may, for example, provide all instructions without taking the user's prior knowledge into account. Alternatively, when the system updates the user's beliefs about known locations, the dialogue controller may decide to first ask if the user recalls an intermediate location (e.g. 'do you remember how to get to the post room?') in order to provide information more efficiently. The third question is addressed by system behaviour that takes the spatial environment into account, such as choosing a route that is easiest for the user to follow or that goes past landmarks that the user is already familiar with. In the wayfinding domain, the vast amount of possible routes to follow prohibits the approach of policy learning in advance, and makes the approach of policy learning in real-time preferable (which demands efficient learning techniques).

3. Learning Situated Dialogue Management

3.1. Learning Dialogue Policies with Hierarchical Control

We treat spatially-aware dialogue control as a discrete Semi-Markov Decision Process (SMDP) in order to address the problem of scalable dialogue optimization. A discrete-time SMDP $M = <S, A, T, R>$ is characterized by a finite set of states S; a finite set of actions A; a stochastic state transition function $T(s', \tau|s, a) = P(s', \tau|s, a)$ that specifies the next state s' given the current state s and action a; and a reward function $R(s', \tau|s, a)$ that specifies the reward given to the agent for choosing action a when the environment makes a transition from state s to state s'. The random variable τ denotes the number of time-steps taken to execute action a in state s. We distinguish two types of actions: (a) single-step actions roughly corresponding to dialogue acts or spatial actions such as 'turn left' or 'turn around', and (b) multi-step actions corresponding to sub-dialogues or contractions of single-step spatial actions such as 'go straight until the end of the corridor'. The solution to a Semi-Markov decision process is an optimal policy π^*, which is a mapping from environment states $s \in S$ to single- or multi-step actions $a \in A$. In addition, we treat each multi-step, spatially-aware dialogue action as a separate SMDP as described in [4, 6]. In this way, an MDP can be decomposed into multiple SMDPs that are hierarchically organized into L levels and N models per level, denoted as $\mathcal{M} = \{M_j^i\}$, where $j \in \{0, ..., N-1\}$ and $i \in \{0, ..., L-1\}$. Thus, a given SMDP in the hierarchy is denoted as $M_j^i = <S_j^i, A_j^i, T_j^i, R_j^i>$. The goal of an SMDP is to find an optimal policy π^* that maximizes the reward of each visited state. The optimal action-value function $Q^*(s, a)$ specifies the expected cumulative reward for executing action a in s and then following π^*. The optimal policy for each model in the hierarchy is defined by $\pi_j^{*i}(s) = \arg\max_{a \in A_j^i} Q_j^{*i}(s, a)$, where the Q-function specifies the cumulative reward for each state-action pair. We use the HSMQ-Learning algorithm [7] for hierarchical policy learning.

3.2. State-Action Spaces with Relational Representations

An MDP is typically represented with propositional representations (e.g. a set of binary features), which result into exponential growth. A relational (Semi-) MDP mitigates that problem by using tree-based and high-level representations resulting in the following benefits: (a) compression and more expressive description of the state-action space, (b) straightforward incorporation of prior-knowledge into the policy, (c) generalization for reusable behaviours, and (d) fast learning. A relational MDP is a generalization of an MDP specified with representations based on a logical language [8]. It can be defined as a 5-tuple $<S, A, T, R, L>$, where element L is a language that provides the mechanism to express logic-based representations. We describe L as a context-free grammar to represent formulas compounded by predicates, variables, constants and connectives similar to [3] (Chapter 8). Whilst the state set S is generated from an enumeration of unique formulas in grammar L, the actions A available in a given state are constrained by the relational representations. A sample relational state is expressed as follows: '$Salutation(greeting) \wedge Slot(x, confirmed) \wedge SlotsToConfirm(none) \wedge SalientLandmarkToAsk(none)$'. This representation indicates that slot x has been confirmed, there are no slots to confirm and no salient landmarks to ask for. A sample relational action is '$request \leftarrow Salutation(greeting) \wedge Slot(x, unfilled)$'. This indicates that the action 'request' is valid if the logical expression is true.

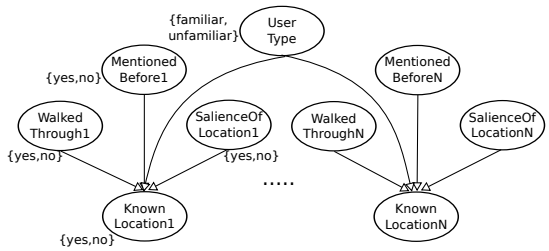

Figure 1: BN for modelling the user's prior knowledge about known locations in the in-situ indoor navigation problem.

3.3. Bayesian Networks for Tracking User Beliefs

The learnt spatial behaviour of the agent is also strongly influenced by an unknown user, for whom we maintain beliefs as he/she interacts with the environment. To achieve this, we use Bayesian networks, which model the dynamics of a set of random variables (e.g. user type, known locations, saliency) and their dependencies. A BN models a joint probability distribution over a set of random variables and their dependencies based on a directed acyclic graph, where each node represents a variable Y_j with parents $pa(Y_j)$ [9]. The Markov condition implies that each variable is only dependent on its parents, resulting in a unique joint probability distribution expressed as $p(Y) = \Pi p(Y_j|pa(Y_j))$, where every variable is associated with a conditional probability distribution $p(Y_j|pa(Y_j))$. Such a network is used for probabilistic reasoning, i.e. the calculation of posterior probabilities given a set of query variable-value pairs. To that end, we use efficient implementations of the variable elimination and junction tree algorithms [10]. The BN's conditional probability tables are updated based on the perceived observations from the environment. After each user query (e.g. How do I get to the copy room?, I'm in front of the lifts), the agent re-learns—using simulation—the dialogue policy using the current user beliefs of the environment.

4. Experimental Setting

Our experiments are based on a spoken dialogue system for in-situ indoor navigation, i.e. route instructions are not given in advance, instead, they are given one by one as the user navigates to the goal. The agent's task is to learn to navigate the user—after each user query—from an origin location to 200 destinations (map available in [5]). The user's task is to ask for route instructions and to navigate each received instruction[1]. For such a purpose, we use a simulated environment derived from a real building that is complex to navigate. We used spatial data derived from a single floor of this building, and represented it as an undirected acyclic graph with 400 equally distributed nodes. This route graph and the stochastic behaviour described in the following subsection form the agent's learning environment.

4.1. The Simulated Environment

The simulated environment has a navigation space with 200 locations (35 are salient), and a simulated user. The latter has four sources of uncertainty: confusions at junctions when navigating to the goal occur 10% of the time, coherent user responses with probability .9 and random otherwise, speech recognition

[1] Sample instruction: Go to the post room first, then facing the main entrance, turn left and go straight until the next corridor on your right.

L := g01 g02 g03 g04 g05 g06 g07 g08 g09 g10 g11 g12 g13 g14 ... g41 g42
g01:= Information(null) $\wedge$ Salutation(none)
g02:= Information(null) $\wedge$ Salutation(greeting)
g03:= Information(collected) $\wedge$ Salutation(greeting)
g04:= Information(presented) $\wedge$ Salutation(greeting) $\wedge$ AnotherQuestion(null)
g05:= Information(presented) $\wedge$ Salutation(greeting) $\wedge$ AnotherQuestion(empty)
g06:= Information(presented) $\wedge$ Salutation(greeting) $\wedge$ AnotherQuestion(yes)
g07:= Information(presented) $\wedge$ Salutation(greeting) $\wedge$ AnotherQuestion(no)
g08:= Information(presented) $\wedge$ Salutation(closing)
g09:= UserOrigin(unfilled) $\wedge$ UserDestination(unfilled) $\wedge$ AUDA(unknown)
g10:= UserOrigin(filled,score) $\wedge$ UserDestination(unfilled) $\wedge$ AUDA(no)
g11:= UserOrigin(unfilled) $\wedge$ UserDestination(filled,score) $\wedge$ AUDA(no)
g12:= UserOrigin O(filled,score) $\wedge$ UserDestination(filled,score) $\wedge$ AUDA(no)
g13:= UserOrigin(filled,score) $\wedge$ UserDestination(unfilled) $\wedge$ AUDA(yes)
g14:= UserOrigin(unfilled) $\wedge$ UserDestination(filled,score) $\wedge$ AUDA(yes)
g15:= UserOrigin(filled,score) $\wedge$ UserDestination(filled,score) $\wedge$ AUDA(yes)
g16:= UserOrigin(confirmed) $\wedge$ UserDestination(unfilled) $\wedge$ AUDA(no)
g17:= UserOrigin(confirmed) $\wedge$ UserDestination(filled,score) $\wedge$ AUDA(no)
g18:= UserOrigin(confirmed) $\wedge$ UserDestination(filled,score) $\wedge$ AUDA(no)
g19:= UserOrigin(filled,score) $\wedge$ UserDestination(confirmed) $\wedge$ AUDA(yes)
g20:= UserOrigin(unfilled) $\wedge$ UserDestination(confirmed) $\wedge$ AUDA(yes)
g21:= UserOrigin(filled,score) $\wedge$ UserDestination(confirmed) $\wedge$ AUDA(yes)
g22:= UserOrigin(confirmed) $\wedge$ UserDestination(confirmed) $\wedge$ AUDA(yes)
g23:= Instructions(unknown) $\wedge$ SalientLandmarkToAsk(none)
g24:= Instructions(unknown) $\wedge$ SalientLandmarkToAsk(one) $\wedge$LKBU(null)
g25:= Instructions(unknown) $\wedge$ SalientLandmarkToAsk(one) $\wedge$ LKBU(empty)
g26:= Instructions(unknown) $\wedge$ SalientLandmarkToAsk(one) $\wedge$ LKBU(yes)
g27:= Instructions(unknown) $\wedge$ SalientLandmarkToAsk(one) $\wedge$ LKBU(no)
g28:= Instructions(known) $\wedge$ InstructionsStatus(nonintroduced)
g29:= Instructions(known) $\wedge$ InstructionsStatus(introduced) $\wedge$ RI(null)
g30:= Instructions(known) $\wedge$ InstructionsStatus(introduced) $\wedge$ RI(empty)
g31:= Instructions(known) $\wedge$ InstructionsStatus(introduced) $\wedge$ RI(yes)
g32:= Instructions(known) $\wedge$ InstructionsStatus(introduced) $\wedge$ RI(no)
g33:= Instructions(known) $\wedge$ InstructionsStatus(provided)
g34:= Location(goal)
g35:= Location(start) $\wedge$ Orientation(orientation) $\wedge$ SLTA(none)
g36:= Location(start) $\wedge$ Orientation(orientation) $\wedge$ SLTA(one)
g37:= Location(corridor) $\wedge$ Corridor(segment) $\wedge$ Orientation(orientation)
g38:= Location(junction) $\wedge$ Junction(graphnode) $\wedge$ Orientation(orientation)
g39:= Location(nextofgoal) $\wedge$ Orientation(orientation)
g40:= Location(other) $\wedge$ Orientation(orientation)
g41:= Location(start) $\wedge$ Orientation(orientation)
g42:= Location(corridor) $\wedge$ Orientation(orientation)
score:= 0.1 | 0.2 | 0.3 |0.4 |0.5 |0.6|0.7 | 0.8 | 0.9 | 1
orientation:= north | south | east | west
graphnode:= junction1 | junction1 | ... | junction35
segment:= corridor1 | corridor2 | ... | corridor82

Figure 2: Context-free grammar defining the language L for the situated indoor wayfinding spoken dialogue system. Abbreviations: AUDA=AmbiguousUserDialogueAct, LKBU= LandmarkKnownByUser, RI=RepeatInstructions, SLTA= SalientLandmarkToAsk , g?=state groups (predicates use variables).

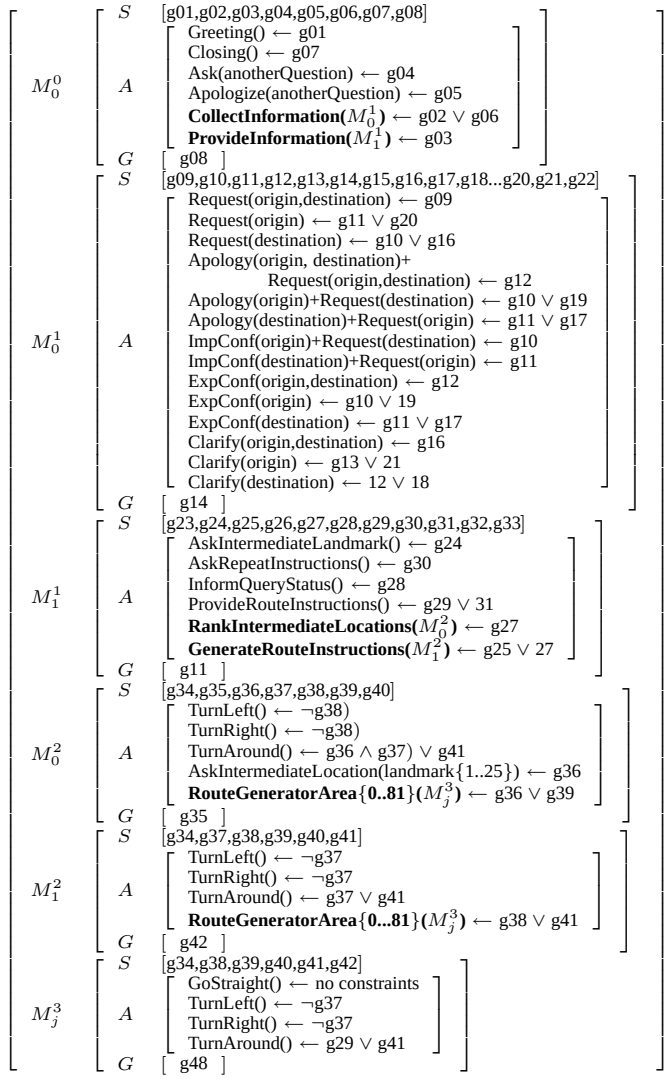

Figure 3: Definition of the Semi-MDPs M_j^i for our situated dialogue agent (Section 4.2 explains these feature structures).

error rate of 20%, and beliefs about known locations. The user beliefs about known salient locations in the environment are modelled with the BN shown in Figure 1. The conditional probability distribution of a user knowing locations is therefore *p(KnownLocation|Salience,WalkedThrough,MentionedBefore, UserType)*. We keep track of this user knowledge so that the agent can optimize its behaviour accordingly. When the user is guided past a landmark, its prior probability of walking through and being mentioned is updated according to $1 - (1/3^v)$, where v is the number of visits. The prior probability of a salient landmark is .9 and .1 otherwise. In addition, the prior probability of the user type being familiar with the environment is updated similarly as the prior probability of walking through a landmark. In this way the simulated user gradually accumulates knowledge of the environment (we assume that a landmark belief is no longer updated if its posterior is $\geq .99$).

4.2. Characterization of the Learning Agent

Figure 2 presents a context-free grammar specifying the semantics of the **dialogue state** for our situated dialogue agent. Dialogue and spatial features are included so as to inform the agent of possible situations in the interaction. Notice that our relational representations abstract away from coordinates and refer to higher level descriptions such as 'corridorX', 'junctionY' and 'nextofgoal'. If we consider each predicate (e.g. Location(x)) in order to specify the state space of our agent with a propositional representation, it corresponds to the order of 10^{11} states. In contrast, our approach based on hierarchical relational representations used only 2875 states. This dramatic compression of the state space shows that our hierarchical relational approach is indeed scalable to larger and more complex dialogue systems.

Figure 3 specifies the **hierarchy of learning agents** with states S, actions A[2] and goal states G, characterized by 87 SMDP models: one parent (M_0^0), two children (M_0^1,M_1^1), two grandchildren (M_0^2,M_1^2), and 82 grand-grandchildren (M_i^3). The latter were induced automatically from turning points in the spatial environment using a random walk approach. Briefly,

[2]Actions can be either primitive (executed within the same SMDP) or composite shown in bold-font (invoke a child SMDP).

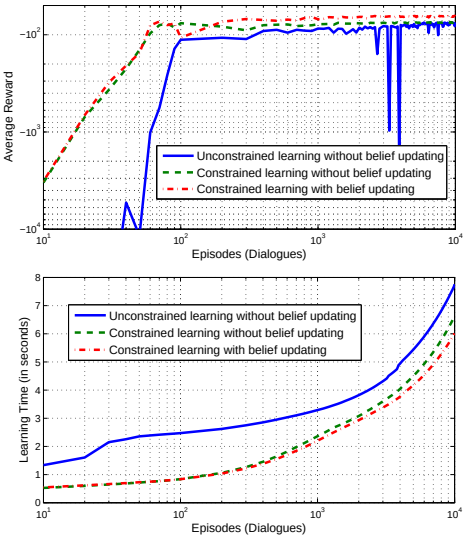

Figure 4: *Learning curves in terms of average rewards (top) and learning time (bottom) of the wayfinding dialogue system.*

the models at the two top layers unify dialogue and spatial behaviour, the models in the following layer provide high-level navigation behaviour (they navigate from one junction to another), and the models in the bottom of the hierarchy provide low-level navigation behaviour (they behave with primitive actions such as 'straight', 'turn left', 'turn right' and 'turn around'). Notice that the **actions** are constrained with the relational states, resulting a set of valid actions for each state.

The **reward function** addressed efficient and effective interactions by penalizing turning instructions more strongly than going straight. The agent receives a reward of 0 for reaching the goal state, -10 for a turning action, -10 for an already executed subtask, -100 for asking for an intermediate location farther than the goal, and -1 otherwise. During training, the learning rate parameter α decays from 1 to 0 according to $\alpha = 100/(100 + \tau)$, where τ represents elapsed timesteps. Each model M_j^i had its own learning rate with undiscounted rewards. The action selection strategy used ϵ-Greedy with $\epsilon = 0.01$, and Q-values initialized to 0.1.

5. Experimental Results

Figure 4 compares three RL agents for six example navigation tasks, each curve averaged over 30 training runs of 10^4 episodes: (a) unconstrained learning (action sets without constraints), (b) constrained learning without belief updating (the user doesn't know about locations), and (c) constrained learning with belief updating (the user accumulates knowledge as he/she navigates to the given destinations). The learning curves are provided according to average reward and learning CPU time in seconds[3]. We can observe that unconstrained learning takes longer to learn a stable behaviour (i.e. more prone to get lost). In contrast, constrained learning finds optimal policies faster and with more stable performance. In addition, we measured the average reward of the last 1000 training dialogues and found that constrained learning with belief updating outperformed its counterpart by an absolute 13% in terms of average reward. The agent with belief updating learned to ask for intermediate locations. Our main experimental result shows that our proposed method is promising for fast policy optimization, for its application in real-time, and therefore suitable for optimizing dialogue behaviour during the course of the interaction with real users. As evidence of the quality of our route instructions, please refer to the human user study reported in [5], where users achieved a binary task success of 93%. The instructions reported were semantically equivalent, even though that system did not track user beliefs, and used propositional state representations.

[3] Experiments used a MacBook Intel Core 2 Duo with 2GB in RAM.

6. Conclusion and Future Work

We have described an approach for optimizing the behaviour of situated dialogue systems, using a combination of HRL with relational representations and BNs. The former was used to optimize dialogue and spatial behaviours entirely in real time, which was possible due to the way that relational representations reduce the agent's search space. BNs were used to track dynamically changing user beliefs during the interaction. This allows flexible and fast adaptation to changing knowledge or arising confusions of the user. The proposed approach makes a contribution to conversational interfaces which learn their dialogue behaviour. These three components (HRL, relational representations and BNs) complemented each other in order to provide optimal guidance: they produced efficient interactions through the use of salient landmarks. Our experimental results provide evidence to conclude that our approach is promising by combining fast learning with adaptive and reasonable behaviour. This research can be extended by (1) an evaluation with real users using spoken interaction with more complex user beliefs, (2) joint optimizations with other behaviours (such as natural language generation and multimodal interaction), and (3) by investigating more complex behaviours in different domains.

7. References

[1] L. Kaelbling, M. Littman, and A. Moore, "Reinforcement learning: A survey," *Journal of AI Research*, vol. 4, pp. 237–285, 1996.

[2] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[4] H. Cuayáhuitl, "Hierarchical reinforcement learning for spoken dialogue systems," Ph.D. dissertation, School of Informatics, University of Edinburgh, 2009.

[5] H. Cuayáhuitl and N. Dethlefs, "Spatially-aware dialogue control using hierarchical rienforcement learning," *ACM Transactions on Speech and Language Processing*, vol. 7, no. 3, pp. 5:1–5:26, 2011.

[6] H. Cuayáhuitl, S. Renals, O. Lemon, and H. Shimodaira, "Evaluation of a hierarchical reinforcement learning spoken dialogue system," *Computer Speech and Language*, vol. 24, no. 2, pp. 395–429, 2010.

[7] T. Dietterich, "An overview of MAXQ hierarchical reinforcement learning," in *Symposium on Abstraction, Reformulation, and Approximation (SARA)*, Jul 2000, pp. 26–44.

[8] M. van Otterlo, *The Logic of Adaptive Behaviour: Knowledge Representation and Algorithms for Adaptive Sequential Decision Making under Uncertainty in First-Order and Relational Domains*. IOS Press, 2009.

[9] F. Jensen, *An Introduction to Bayesian Networks*. Springer Verlag, New York, 1996.

[10] F. G. Cozman, "Generalizing variable elimination in Bayesian networks," in *IBERAMIA/SBIA, Workshop on Probabilistic Reasoning in Artificial Intelligence*, 2000, pp. 27–32.