Spatially-Aware Dialogue Control Using Hierarchical Reinforcement Learning

HERIBERTO CUAYÁHUITL and NINA DETHLEFS, University of Bremen

This article addresses the problem of scalable optimization for spatially-aware dialogue systems. These kinds of systems must perceive, reason, and act about the spatial environment where they are embedded. We formulate the problem in terms of Semi-Markov Decision Processes and propose a hierarchical reinforcement learning approach to optimize subbehaviors rather than full behaviors. Because of the vast number of policies that are required to control the interaction in a dynamic environment (e.g., a dialogue system assisting a user to navigate in a building from one location to another), our learning approach is based on two stages: (a) the first stage learns low-level behavior, in advance; and (b) the second stage learns high-level behavior, in real time. For such a purpose we extend an existing algorithm in the literature of reinforcement learning in order to support reusable policies and therefore to perform fast learning. We argue that our learning approach makes the problem feasible, and we report on a novel reinforcement learning dialogue system that performs a joint optimization between dialogue and spatial behaviors. Our experiments, using simulated and real environments, are based on a text-based dialogue system for indoor navigation. Experimental results in a realistic environment reported an overall user satisfaction result of 89%, which suggests that our proposed approach is attractive for its application in real interactions as it combines fast learning with adaptive and reasonable behavior.

Categories and Subject Descriptors: H.5.2 [**User Interfaces**]: Natural Language—*Dialogue systems*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Conversational agents*

General Terms: Algorithms, Performances, Experimentation

Additional Key Words and Phrases: Dialogue systems, machine learning, spatial cognition, dialogue optimization, reinforcement learning, route instruction generation, hierarchical control, policy reuse, dynamic environments, system evaluation

ACM Reference Format:
Cuayáhuitl, H. and Dethlefs, N. 2011. Spatially-aware dialogue control using hierarchical reinforcement learning. ACM Trans. Speech Lang. Process. 7, 3, Article 5 (May 2011), 26 pages.
DOI = 10.1145/1966407.1966410 http://doi.acm.org/10.1145/1966407.1966410

1. INTRODUCTION

Reinforcement learning may be used to infer optimal behaviors for conversational interfaces. A reinforcement learning agent learns its behavior from interaction with an environment, where situations are mapped to actions by maximizing a long-term reward signal. The standard reinforcement learning paradigm works under the formalism of Markov Decision Processes (MDPs) [Kaelbling et al. 1996; Sutton and Barto 1998; Russell and Norvig 2003]. An MDP is characterized by a finite set of states S (corresponding to situations in the dialogue), a finite set of actions A (corresponding to

Part of this work was sponsored by the Transregional Collaborative Research Center SFB/TRS Spatial Cognition. Funding by the German Research Foundation (DFG) is gratefully acknowledged.
Author's address: H. Cuayáhuitl and N. Dethlefs, Spatial Cognition Research Center SFB/TRS, Enrique Schmidt Strasse 5, 28358, Bremen, Germany; email: hcuayahu@uni-bremen.de.

DOI 10.1145/1966407.1966410 http://doi.acm.org/10.1145/1966407.1966410

dialogue actions), an unknown state transition function, and a reward or performance function that rewards the agent for each selected action. Solving the MDP means finding a mapping from the current state s_t to an action a_t corresponding to a dialogue policy $\pi^*(s_t) = \arg\max_{a_t \in A} Q^*(s_t, a_t)$, where the Q-function specifies the cumulative rewards for each state-action pair.

In the past, dialogue systems that learn to optimize their behavior have typically been investigated using flat tabular reinforcement learning [Levin et al. 2000; Walker 2000; Young 2000; Singh et al. 2002; Scheffler 2002; Pietquin 2004]. The scalability of this approach is limited because state spaces grow exponentially according to the number of state variables taken into account. This problem, referred to as *the curse of dimensionality*, has been addressed in different ways: (a) by function approximation techniques [Denecke et al. 2004; Henderson et al. 2008], which find solutions on reduced search spaces; (b) by evolutionary methods for learning dialogue strategies that mitigate the size of search spaces [Toney 2007]; and (c) by using master, summary, and factorized spaces [Williams 2006; Thomson 2009; Young et al. 2010]. These investigations have been applied to dialogue systems resulting in a single global solution, and less attention has been paid to finding solutions using divide-and-conquer approaches [Cuayáhuitl et al. 2010c; Lemon 2010]. The latter have a number of benefits: their aim is not only to scale a single optimization module such as dialogue management, but they also aim at jointly optimizing different modules such as dialogue management and language generation.

This paper focuses its attention on reinforcement learning for spatially-aware dialogue systems, which must perceive, reason, and act in relation to the environment in which they are embedded. In such systems the dialogue behavior is strongly influenced by the particular domain in which they behave. We focus on the wayfinding domain, and argue that dialogue management and route generation behavior need to be optimized in a unified way in order to address adaptive behavior within a spatial environment. In this way, the dialogue manager can draw on spatial knowledge to choose optimal dialogue actions, and the route generator can derive optimal route instructions, given a dynamic spatial environment. Furthermore, it will be necessary for learning to occur in real time (while) using a simulator (also referred to as real-time AI [Russell and Norvig 2003]).

Consider the following two scenarios: (a) dialogue behavior weakly coupled with spatial behavior, where dialogue actions are independent of spatial ones; and (b) dialogue behavior tightly-coupled with spatial behavior, where dialogue actions depend on spatial ones. While *offline* learning may be sufficient for the former scenario, only *online* learning (applied when the environment changes, for instance after a user's query) is suitable for the latter scenario. Moreover, the environment is continuously changing (because of either new user prior knowledge, new world objects, or constraints in the navigation space), and the system has to behave accordingly. This tells us that offline learning cannot be applied to spatially-aware conversational interfaces. Since the spatial environment is constantly changing, situated dialogues with adaptive behavior demand efficient learning techniques. We are not aware of any related work using such a kind of learning for dialogue systems.

It has to be remembered that even though the dialogue system we present optimizes behavior specifically for the wayfinding domain, our proposed approach is more general and can be transferred to different domains. The work that relates most closely to ours is Lemon [2010], which presents an approach for unified optimization of dialogue management and language generation for content selection and information presentation strategies in the domain of restaurant and music recommendations. His results showed that a system that learned behavior in a unified fashion can outperform a system that learned both policies in isolation.

In the rest of the article we show that reinforcement learning with a hierarchical setting and policy reuse is particularly relevant to the situated domain. This is because it merges dialogue behavior with spatial behavior into a unified, efficient, and scalable optimized behavior. The article is organized as follows. Section 2 introduces findings from spatial cognition research that are important in informing the behavior of a situated wayfinding assistant. Section 3 introduces reinforcement learning for spatially-aware dialogue systems with a hierarchical setting. Sections 4 and 5 present the results we obtained with our proposed approach. The former section describes results obtained with a simulated environment; the latter section reports an evaluation in a real setting with human participants. We then provide a discussion of our work contrasted with the current literature in Section 6. Finally, Section 7 draws conclusions and comments on avenues for future research.

2. TOWARDS SPATIALLY-AWARE DIALOGUE SYSTEMS

The behavior of situated spoken dialogue systems is strongly influenced by the particular domain in which they operate, in our particular case the wayfinding domain. The challenge that arises with the wayfinding domain in particular is that information can be presented to a user in many different ways. First, there can be multiple routes from an origin to a destination, for instance, the easiest route to follow, the shortest route, the route simplest to describe, etc., and it is not trivial to decide which route is the best, given the current user and the properties of the current spatial environment (including the complexity of junctions, salience of landmarks, or length of the route). Second, routes can be provided at different degrees of granularity, the best level will depend on the users' knowledge of the environment and the complexity of the route. Finally, there are different ways of realizing the surface form of routes, including the realization as a full text, or (on the opposite end of abstraction) a list of schematic instructions, or several variants between these two extremes. 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 navigation environment—the experience of navigating the environment that users bring with them, but also knowledge about the spatial environment in which the system navigates. In addition, empirical findings on the principles that typically guide human wayfinding behavior can help make route instructions more easily comprehensible.

2.1. Human Wayfinding Behavior

Based on studies on human route descriptions, Lovelace et al. [1999] suggested the following as characteristics of good route descriptions, which can be used by wayfinding dialogue systems: (a) provide look-ahead information to prepare the user for upcoming choice points, (b) include salient landmarks to provide additional points of orientation, (c) provide confirmatory information to assure users they are still on the right track, (d) present information in a sequential way corresponding to the sequence of actions to be taken, (e) avoid redundancy, and (f) avoid metrical distances because humans find it difficult to estimate these exactly.

In addition to these features, principles of spatial chunking and adaptation to spatial environments and user groups have been assumed to enhance the cognitive adequacy of route instructions. Landmarks definitely appear to play a key role. They are highly prominent features in route descriptions [Denis 1997], because they help humans structure their knowledge of an environment [Sorrows and Hirtle 1999]. As a further crucial feature, spatial chunking provides a means of transforming small route segments (that may directly correspond to the output of some routing algorithm) into high-level instructions that humans may find easier to process and remember [Klippel et al. 2009]. Humans also tend to adapt their instructions to changing spatial situations, such as

the type or complexity of intersections, the presence or absence of salient landmarks, or the actions that need to be performed at various points [Klippel et al. 2010]. Finally, there is adaptation to the information needs of different user groups in human route descriptions, which take into account users' means of transportation [Tenbrink and Winter 2009] or prior knowledge [May et al. 2003; Burnett et al. 2001].

2.2. Spatial Knowledge in a Wayfinding Dialogue System

The following elements can be taken into account by wayfinding dialogue systems.

—*Spatial data.* Knowledge of a given spatial environment can be specified as a map, for instance, by using geometric and semantic information represented as ontologies. The former type of information specifies generic elements such as points, lines, polylines and polygons. The latter specifies real world objects (e.g., rooms, corridors, walls, doors) and a route graph that models the navigational space. The route graph consists of a set of points and segments connecting such points [Werner et al. 2000]. In addition, the world objects have a set of features such as names, colors, owners, etc. The prominent objects are referred to as landmarks.
—*Ranking of landmarks.* We follow Raubal and Winter [2002] in ranking landmarks based on the categories suggested by Sorrows and Hirtle [1999]. For this purpose, we identified the relevant types of landmarks in our domain, such as offices, toilets, or classrooms, and assigned them a weight between 0 and 1 indicating their relative salience in the environment. In this way, we can anticipate both the appropriateness of a landmark for inclusion in an instruction as well as the likelihood with which a user will be familiar with the location.
—*Confusion probabilities of junctions.* In a similar fashion, junctions can be assigned a weight depending on their complexity, taking into account the number of paths leaving a junction, or the salience weights of present landmarks. The assigned weights correspond to the likelihood with which users may get lost at this point. This information can be used, on the one hand, for route planning in that the easiest route will be the one with least likelihood of getting lost. On the other hand, it can be used for route descriptions. Whenever a junction with high confusion probability needs to be visited by the user, the system can decide to include additional detail and thereby correspond to the user's need for information.
—*Interactively guided wayfinding.* The sample dialogues presented in Table I show evidence that a situated dialogue manager must perform reasoning about the spatial environment. In particular, a spatially-aware dialogue manager must deal with the following questions:
  —When to present information?
  —When to ask for users' prior knowledge?
  —What information to present according to the dialogue history and spatial environment?

  We explain these questions with the dialogues shown in Table I. Consider the following two scenarios: (a) the dialogue manager provides all necessary instructions without taking into account the user's prior knowledge, see first dialogue; and (b) the dialogue manager decides first to ask if the user knows the intermediate location (or landmark) "stuga room" and then provide information, see second dialogue. These decisions depend on several factors, such as the length of the paths to the goal and the prior knowledge of known landmarks along these paths. In this way, they are very peculiar to the spatial setting and cannot be optimally addressed without a tight coupling of spatial information and dialogue management. The third question is addressed by system behavior that takes the spatial environment into

Table I.
Sample dialogues in the wayfinding domain showing in-advance indoor route instructions, where the latter system provides shorter route instructions due to a known intermediate landmark. Landmarks refer to reference relations such as $\{r_1 = \text{after}, r_2 = \text{before}, r_3 = \text{right}, r_4 = \text{left}, r_5 = \text{front}\}$

Agent	Dialogue Act	Utterance
Sys^1	greeting()	Welcome to the wayfinding info-kiosk.
	request(slot=destination)	How can I help you?
Usr		*How do I get to room A3445?*
Sys^1	generateRouteInstructions(origin=default, destination=a3445)	
	informQueryStatus()	The following are your route instructions.
	verbalizeInstructions(<turn around, straight,next corridor,glass door:r_1>	Turn around, then go straight on, passing the glass door, until the corridor. Turn left, and go straight
	<left,straight,stuga room,message board:r_3>	on, passing the message board on your right, until the stuga room.
	<right,straight,next corridor,glass door:r_1>	Turn to the right, then go straight, passing the glass door, until the end of the corridor.
	<right,straight,door A3445:r_3>)	Turn to the right, then go straight until the door A3445 on your right.
	askRepeatInstructions()	Do you want to hear that again?
Usr		*No thanks*
Sys^1	askAnotherQuestion()	Do you have another question?
Usr		*No, I don't*
Sys^1	closing()	Thanks for using the wayfinding info-kiosk. Good bye!
Sys^2	greeting()	Welcome to the wayfinding info-kiosk.
	request(slot=destination)	How can I help you?
Usr		*How do I get to Dr. Watson's office?*
Sys^2	clarify(entries={james watson,peter watson})	Do you mean James or Peter Watson?
Usr		*Peter Watson*
Sys^2	rankIntermediateLocations(origin=default,destination=peter watson)	
	askIntermediateLocation(landmark=stuga room)	Do you know how to get to the stuga room?
Usr		*Yes I do*
Sys^2	generateRouteInstructions(origin=stuga room, destination=peter watson)	
	informQueryStatus()	OK, the following are your route instructions.
	verbalizeInstructions(<stuga room,go,main entrance:r_5> <left,straight,next corridor:r_3>	Go to the stuga room first, then facing the main entrance, turn left and go straight until the next corridor on your right.
	<right,straight,door A3445:r_3>)	Turn right and go straight until door A3445 that will be on your right.
	askRepeatInstructions()	Do you want to hear that again?
Usr		*No thanks*
Sys^2	askAnotherQuestion()	Do you have another question?
Usr		*No, good bye*
Sys^2	closing()	Thanks for using the wayfinding info-kiosk. Good bye!

account, such as choosing a route that is easiest for the user to follow or that goes past landmarks with which the user is familiar.

In this article we argue that the behavior of situated dialogue managers is not easy to hand-code and therefore needs to adopt optimization methods. Furthermore, reinforcement learning for our domain is challenging because it has to occur in real time, during the course of the interaction. The rest of the article illustrates our approach to this problem.

3. REINFORCEMENT LEARNING FOR SITUATED INTERACTION

Typically, learned dialogue behavior is induced in advance, before interacting with real users. This approach is problematic for a number of reasons. First, dialogue should take into account users' prior knowledge, for instance, Do you know how to get to the post room? Second, content selection should be optimized according to a changing spatial environment, for instance, What route to follow? What landmarks to include? Third, spatially-aware behavior needs to be estimated online because of the vast number of policies. This is justified by the fact that $\sum_{r>1}^{n} n!/(n-r)!$ unique routes are possible for n queryable locations of length r, assuming that wayfinding can occur from any location to any other in the space. Even when route graphs are usually not fully connected, this vast amount of possible routes prohibits the approach of policy learning in advance, and makes the approach of policy learning in real time preferable. The rest of this section shows how to tackle this problem.

3.1. Background on Reinforcement Learning Dialogue Agents

A reinforcement learning agent senses and acts in its environment in order to learn to select optimal actions to achieve its goal. The agent's task is to learn a policy or control strategy for choosing the best actions in the long run that will achieve its goal. For such a purpose the agent maintains a cumulative reward for each state or state-action pair. The environment is usually represented as a Markov Decision Process (MDP) or as a Partially Observable MDP (POMDP). In this paper we focus our attention on the former model. A reinforcement learning agent interacting with an environment described as a Markov Decision Process is defined as a 4-tuple $\langle S, A, T, R\rangle$ characterized as follows.

—S is a finite set of states in the environment, where $S = \{s_0, s_1, \ldots, s_N\}$ and s_t is the state at time t. The states in an MDP are directly observable, used to describe all possible situations in the spatial environment, and the basis for action selection. Assuming that we cast dialogue optimization as an episodic task, then the state set includes nonterminal states and terminal states. The state at time $t+1$ is also denoted as s'.
—A is a finite set of actions available in the spatial environment, where $A = \{a_0, a_1, \ldots, a_M\}$ and a_t is the action at time t. When action a_t is executed, it changes the current state of the world from s_t to s_{t+1}. The action at time $t+1$ is also denoted as a'.
—T is a state transition function that observes the next state s' given the current state s and action a. This state transition function is represented with a conditional probability distribution $P(s'|s,a)$ satisfying $\sum_{s'\in S} P(s'|s,a) = 1, \forall(s,a)$.
—R is the reward function that specifies the immediate reward r_t at time t given to the agent for choosing action a when the environment makes a transition from s to s'. The reward at time $t+1$ is also denoted as r'.

To control a system described as a Markov Decision Process, one needs a decision-making function or policy π, which is a mapping from environment states $s \in S$ to actions $a \in A$. A stochastic policy is denoted as $\pi(s) = P(a|s)$ and a deterministic policy is denoted as $\pi(s) = a$. The optimal solution for an MDP is that of taking the best action a_t available in state s_t, that is, the action that collected as much reward as possible over

time. A given sequence of states, actions, and rewards $\{s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2 \ldots\}$, receives a total cumulative discounted reward expressed as $r = r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots \gamma^{\tau-1} r_\tau = \sum_{k=0}^{\tau-1} \gamma^k r_{k+1}$, where the discount rate $0 \leq \gamma \leq 1$ makes future rewards less valuable than immediate rewards as it approaches 0. An *optimal policy* performs action selection according to

$$\pi^*(s) = \arg\max_{a \in A} Q^*(s, a), \tag{1}$$

where the Q-function specifies the cumulative reward of starting in state s, taking action a and then following policy π^* thereafter. The optimal policy can be learned by reinforcement learning methods such as Q-Learning or SARSA. For example, Q-Learning computes Q-values according to

$$Q(s, a) \leftarrow Q(s, a) + \alpha\Big[r + \gamma \max_{a' \in A} Q(s', a') - Q(s, a)\Big]. \tag{2}$$

Q-Learning updates values for sample state-action pairs (s, a), where the execution of action a in state s yields state s' and reward r. γ is a discount rate in the range $[0, 1]$, and α is a learning rate parameter that decays from 1 to 0. An important characteristic of reinforcement learning is the trade-off between exploration and exploitation. The agent has to perform *exploration* in order to discover better behaviors, but it also has to perform *exploitation* of the already learned behavior in order to obtain more reward. In this dilemma, a learning agent must try different actions and progressively prefer those that seem to be the best. A reinforcement learning agent can perform exploitation (1) with a fixed probability, a method referred to as ϵ-greedy action selection; (2) according to a probability distribution of cumulative rewards $Q(s, a)$, also referred to as *softmax* action selection; or (3) the learning method (such as policy-gradient or Bayesian methods) can optimize this trade-off to reduce large changes in the value function.

Although MDP-based reinforcement learning offers an attractive framework for optimizing the behavior of conversational systems, its practical application is affected by the following problems: the curse of dimensionality, partial observability, and learning from real interactions. In the first, the state space growth is exponential in the number of state variables. In the second, the dialogue agent operates under uncertainty (e.g., automatic speech recognition errors). In the third, reinforcement learning methods require a large number of dialogues to find optimal policies. These problems offer motives for proposing alternative optimization approaches. In the rest of this section we tackle the first and last problem using a hierarchical approach and show how to apply it to spatially-aware dialogue systems.

3.2. Spatially-Aware Dialogue Control Using Hierarchical SMDPs

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 $M = \langle S, A, T, R \rangle$, as formulated by Dietterich [2000a], is characterized by a set of states S; a set of actions A; a transition function T that specifies the next state s' given the current state s and action a with probability $P(s', \tau | a, s)$; 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. The SMDP model allows temporal abstraction, where actions take a variable amount of time to complete their execution. In this model two types of actions can be distinguished: (a) single-step actions roughly corresponding to dialogue acts or spatial actions such as "turn left" or "turn around", and (b) multistep actions corresponding to subdialogues or

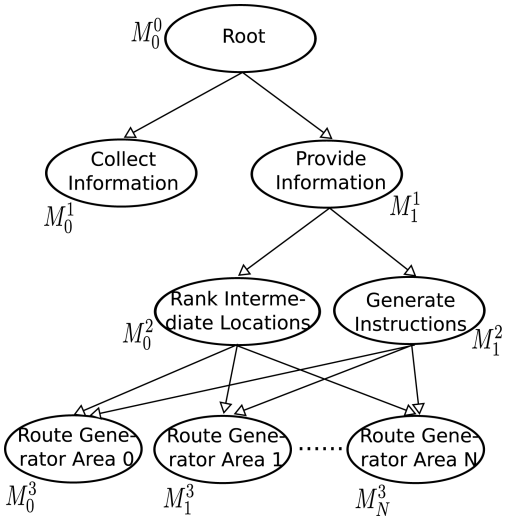

Fig. 1. Top-down hierarchy of reinforcement learning agents for wayfinding dialogue systems, representing high-level behaviors and encapsulating lower-level ones. When a child agent terminates its execution, control is transfered to its parent and so on until the end of the dialogue.

spatial actions such as "go straight until the end of the corridor" (consisting of N single-step actions such as "straight, straight, straight, half left, straight, straight, straight"). The solution to a Semi-Markov Decision Process is an optimal policy π^*, which is a mapping from environment states $s \in S$ to primitive or composite actions $a \in A$.

This research treats each composite, spatially-aware dialogue action as a separate SMDP as suggested in [Cuayáhuitl et al. 2007; Cuayáhuitl 2009; Cuayáhuitl et al. 2010c]. In this way a Markov Decision Process (MDP) can be decomposed into multiple Semi-MDPs that are hierarchically organized into L levels and N models per level, denoted as $\mathcal{M} = \{M_j^i\}$, where $j \in \{0, \ldots, N-1\}$ and $i \in \{0, \ldots, L-1\}$. Thus, a given SMDP in the hierarchy is denoted as $M_j^i = \langle S_j^i, A_j^i, T_j^i, R_j^i \rangle$. 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 Bellman equation for Q^* of model (also referred to as subtask) M_j^i can be expressed as

$$Q_j^{*i}(s,a) = \sum_{s',\tau} P_j^i(s',\tau|s,a)\big[R_j^i(s',\tau|s,a) + \gamma^\tau \max_{a'} Q_j^{*i}(s',a')\big]. \quad (3)$$

Finally, the optimal policy for each model in the hierarchy is defined by

$$\pi_j^{*i}(s) = \arg\max_{a \in A_j^i} Q_j^{*i}(s,a). \quad (4)$$

These policies can be found using the reinforcement learning algorithms described in the next subsection (also applicable to dialogue systems in other domains). Before that, we briefly address the following question: How should a spatially-aware MDP be decomposed into subproblems? Because the process of automatically breaking an MDP into subproblems is challenging, heuristic approaches can be used to manually decompose such tasks. The general idea is to decompose a dialogue into subdialogues, and to decompose a spatial task into smaller tasks. Our hierarchy for a wayfinding dialogue system is shown in Figure 1. We used a handcrafted hierarchical dialogue structure and a hierarchical spatial structure induced from the spatial data (corresponding

to models M_j^2 and M_j^3 in the hierarchy above). This hierarchy shows dialogue subtasks such as "collect information" and spatially-aware subtasks such as "provide information".

3.3. Hierarchical Reinforcement Learning Algorithms

The *HSMQ-Learning algorithm* (Algorithm 1) simultaneously learns a hierarchy of SMDP-based action-value functions $Q_j^{*i}(s,a)$ [Dietterich 2000b]. This algorithm has been used to optimize the behavior of information-seeking dialogue systems [Cuayáhuitl 2009; Cuayáhuitl et al. 2010c]. Briefly, this learning algorithm receives dialogue subtask M_j^i, and knowledge base[1] k used to initialize state s. It performs similarly to Q-Learning for primitive actions, but for composite actions it invokes recursively with a child subtask. The execution of subtasks uses a stack and operates as follows: the dialogue starts with the root subtask M_0^0 on the stack; when a child subtask M_j^1 is selected, it is pushed onto the stack and control is transferred to the child subtask which is executed until reaching a terminal state—this may involve a recursive execution of other subtasks that may reach the bottom of the hierarchy; then the current subtask is popped off the stack and control is transferred back to the parent subtask at the next state $s' \in S_j^i$; this process continues until the execution of the root subtask is completed, which empties the stack and terminates the dialogue. When a given subtask is completed with τ time steps, it returns a cumulative reward $r_{t+1}+\gamma r_{t+2}+\gamma^2 r_{t+3}+\cdots+\gamma^{\tau-1} r_{t+\tau}$, and continues its execution until finding a terminal state for the root subtask M_0^0. This algorithm is iterated until convergence occurs to optimal context-independent policies.[2]

The HSMQ-Learning algorithm just described (though applicable to information-seeking systems) has limited applicability to situated dialogue systems mainly due to slow learning. Because spatially-aware dialogue control requires learning in real-time, here we show an extension of the HSMQ-Learning algorithm for more efficient learning. The *HSMQ-Learning algorithm with policy reuse* shown in the following uses a two-stage approach. The first stage, applied before user-machine interaction, induces policies from a stationary simulation environment with static goal states (e.g. purely dialogic policies such as collect information, or policies at the bottom of the hierarchy that do not include the user's goal location). The second stage, applied in real time after a human request, induces policies from a nonstationary simulation environment with dynamic goal states (e.g., agents referring to a goal location and parent policies). The difference between static and dynamic goal states is that the latter refer to unknown locations; for instance, while one user navigates to location X, another user navigates to location Y, and so on. While the first stage applies the HSMQ-Learning algorithm to each subtask, the second stage applies the HSMQ-Learning algorithm with policy reuse in order to learn a unified hierarchical policy. This approach avoids learning from scratch by reusing learned behavior from the first stage in order to focus on learning high-level behavior, and therefore to infer policies with fast learning. For example, going back to the hierarchy of agents for the dialogue system shown in Figure 1, while most of the leaf agents are part of the first stage, nonleaf agents are part of the second stage. In this

[1]The knowledge base keeps track of all the information generated through the dialogue history by holding attribute-value pairs represented in an ontology-based structure.

[2]An optimal context-independent policy achieves the highest cumulative reward for the given composite action, but suffers from being locally optimal rather than globally optimal. Here, temporally extended behaviors execute actions that are locally optimal. The advantage is that context-independent policies facilitate state abstraction and policy reuse, but context-dependent policies allow stronger optimality. The latter are weaker for state abstraction and policy reuse.

ALGORITHM 1: HSMQ-Learning for dialogue control

function HSMQ(KnowledgeBase k, subtask M_j^i) **return** $totalReward$
  $s \leftarrow$ knowledge-compact state in S_j^i initialized from k
  $totalReward \leftarrow 0$, $discount \leftarrow 1$
  while s is not a terminal state **do**
    Choose action a from s using policy derived from Q_j^i (e.g. ϵ-greedy)
    Execute action a and update knowledge-rich state k
    if a is primitive **then**
      Observe one-step reward r
    else if a is composite **then**
      $r \leftarrow$ HSMQ(k, model of composite action a)
    end if
    $totalReward \leftarrow totalReward + discount \times r$
    $discount \leftarrow discount \times \gamma$
    Observe resulting state s'
    $Q_j^i(s,a) \leftarrow (1-\alpha)Q_j^i(s,a) + \alpha\left[r + discount \times \max_{a' \in A_j^i} Q_j^i(s',a')\right]$
    $s \leftarrow s'$
  end while
end function

example, the SMDP models that require learning in real time are $\{M_1^1, M_0^2, M_1^2, M_j^3\}$, where the latter represents a subset of models depending on the number of routes that lead to the goal location. These models determine the route to follow and whether or not to ask for intermediate locations. Note that when a child agent (re-) learns its behavior, its parent also relearns its behavior in order to maintain optimal behavior. Algorithm 2 is also iterated until convergence occurs to optimal context-independent policies [Dietterich 2000a; Dietterich 2000b]. Although both previous algorithms can be applied to dialogue systems in different domains, a key advantage of the latter algorithm is its application to scenarios that require learning in real time.

4. EXPERIMENTS USING A SIMULATED ENVIRONMENT

The aim of our experiments was threefold: (1) to show that hierarchical reinforcement learning is suitable for joint optimization of dialogue and spatial behaviors, (2) to investigate the potential application of our proposed learning approach by comparing HSMQ-Learning with and without policy reuse, and (3) to induce adaptive behavior for familiar and unfamiliar users.

We hypothesize that in this way: (a) the joint optimization of dialogue and spatial behavior leads to better performance than optimizing the two in isolation; (b) HSMQ-Learning with policy reuse leads to quicker learning than without policy reuse and thereby makes our approach suitable for spatially-aware dialogue control; and that (c) the system will learn to adapt its behavior to users' prior knowledge by asking familiar users for intermediate landmarks—if any suitable candidates are present in the environment—and providing full instructions to unfamiliar users.

Our experiments are based on a dialogue system for indoor navigation. Our reinforcement learning agents learned their behavior from a simulated environment with spatial data derived from a real building that is complex to navigate.[3] We used data from a single floor (see Figure 4), and represented it as an undirected acyclic graph with

[3]Our dialogue system can be seen as two versions, one for each type of user. The system does not induce the user type online, instead, it is run for one user type because interactions are short and do not provide enough information for inducing the user type online. We left the induction of the user type as future work,

ALGORITHM 2: HSMQ-Learning with policy reuse for dialogue control

function HSMQ_PR(KnowledgeBase k, subtask M_j^i, boolean *reusable*) **return** $totalReward$
  $s \leftarrow$ knowledge-compact state in S_j^i initialized from k
  $totalReward \leftarrow 0$, $discount \leftarrow 1$
  while s is not a terminal state **do**
    if M_j^i is reusable **then**
      Choose action a from $\pi_j^{*i}(s) = \arg\max_{a \in A_j^i} Q_j^{*i}(s,a)$
    else
      Choose action a from s using policy derived from Q_j^i (e.g. ϵ-greedy)
    end if
    Execute action a and update knowledge-rich state k
    if a is primitive **then**
      Observe one-step reward r
    else if a is composite **then**
      reusable $\leftarrow$ false if action a has dynamic goal states, or if the children of
        action a have (re-) learnt their behavior; true otherwise
      $M_j^i \leftarrow$ model of composite action a, with dynamic goal states updated
      $r \leftarrow$ HSMQ_PR(k, M_j^i, *reusable*)
    end if
    $totalReward \leftarrow totalReward + discount \times r$
    $discount \leftarrow discount \times \gamma$
    Observe resulting state s'
    if M_j^i is not reusable **then**
      $Q_j^i(s,a) \leftarrow (1-\alpha) Q_j^i(s,a) + \alpha \left[r + discount \times \max_{a' \in A_j^i} Q_j^i(s',a') \right]$
    end if
    $s \leftarrow s'$
  end while
end function

400 equally distributed nodes. This route graph and the stochastic behavior shown in Subsection 4.2 form the agent's learning environment.

4.1. Characterization of the Learning Agent

The learning agent used the state-action space shown in Tables II and III, which results in a space of $|S \times A| = 1.18 \times 10^8$ state-actions. Because tabular flat reinforcement learning is not feasible for this application, we divided the state-action space into the hierarchy shown in Figure 1. Here for illustration purposes, we focus on the right branch because it involves a mixture of dialogue and spatial behavior. The characterization of our hierarchical learning agent used 85 models (see Figure 2): one parent, two children, and 82 grandchildren. The latter were induced automatically from turning points in the spatial data, for which we used a random walk approach. Briefly, the model at the top unifies dialogue and spatial behavior, the models in the middle of the hierarchy provide high-level navigation behavior (they navigate from one junction to another), and the models in the bottom of the hierarchy provide low-level navigation behavior (they behave with primitive actions). In contrast to the flat state-action space representation, our hierarchical representation used only $200K$ state-actions. The state transitions involved a stochastic environment described in the next section.

where longer interactions such as in-situ route instructions provide more information so that the system can induce the user type during the course of the interaction.

Table II.
State variables for the wayfinding dialogue system. Considering a graph with 400 nodes, the state space corresponds to $|S| = 3 \times 4 \times 4 \times 3 \times 2 \times 400 \times 4 \times 2 \times 2 \times 2 = 3.7 \times 10^6$ states

Variable	Values	Description
Instructions	{0, 1, 2}	Corresponding to 'unknown', 'known', 'provided'
KnownIntermediateLandmark	{0, 1, 2, 3}	Corresponding to 'null', 'empty', 'yes', 'no'
RepeatInstruction	{0, 1, 2, 3}	Corresponding to 'null', 'empty', 'yes', 'no'
SalientLandmarkToAsk	{0, 1, 2}	Corresponding to 'unknown', 'none', 'known'
StatusProvideInformation	{0, 1}	Corresponding to 'unknown', 'informed'
Location	(x, y)	Discrete coordinates of graph nodes from the space
Orientation	{0, 1, 2, 3}	Corresponding to 'south', 'north', 'east', 'west'
SalientLandmark	{0, 1}	Corresponding to 'absent', 'present'
SegmentLength	{0, 1}	Corresponding to 'short' and 'long'
UserType	{0, 1}	Corresponding to 'unfamiliar' and 'familiar'

Table III. Set of Primitive Actions ($|A| = 32$) for the Wayfinding Dialogue System

Action	Description
askIntermediateLocation{0...24}	Ask for an intermediate location out of 25 best known
informQueryStatus	Inform status of found or not found location
verbalizeInstructions	Provide route instructions
askRepeatInstructions	Ask for repetition
goStraight	Go straight
turnLeft	Turn left
turnRight	Turn right
turnAround	Turn around

The reward function—used by all models in the hierarchy—addresses efficient interactions by penalizing more strongly turning instructions than going straight. It is defined by the following rewards given to the learning agent for choosing action a when the environment makes a transition from state s to next state s':

$$r(s, a, s') = \begin{cases} 0 & \text{for reaching the goal state} \\ -5 & \text{for a turning action} \\ -10 & \text{for an already executed subtask or asked location} \\ -100 & \text{for asking for an intermediate location farther than the goal} \\ -1 & \text{otherwise.} \end{cases} \quad (5)$$

The learning setup used the learning algorithms described in the previous section (HSMQ-Learning with and without policy reuse). The learning parameters were the same for both learning algorithms. The learning rate parameter α decays from 1 to 0 according to $\alpha = 100/(100 + \tau)$, where τ represents elapsed time-steps in the current subtask. Each subtask M_j^i had its own learning rate with undiscounted rewards. The action selection strategy used ϵ-Greedy with $\epsilon = 0.01$, and initial Q-values of 0.01. We used optimistic initial Q-values in order to encourage exploration at the beginning, with the aim of accelerating the stabilization of learning curves.

4.2. The Simulated Environment

We used a simulated user to navigate in the stochastic environment taking into account two sources of uncertainty. First, the user has confusions when navigating to the goal stemming from complex junctions or the absence of salient landmarks for anchoring the choice points in the environment. Confusions are expressed by the conditional probability

$$P(Confusion|Location, Orientation, SalientLandmark, SegmentLength, UserType), \quad (6)$$

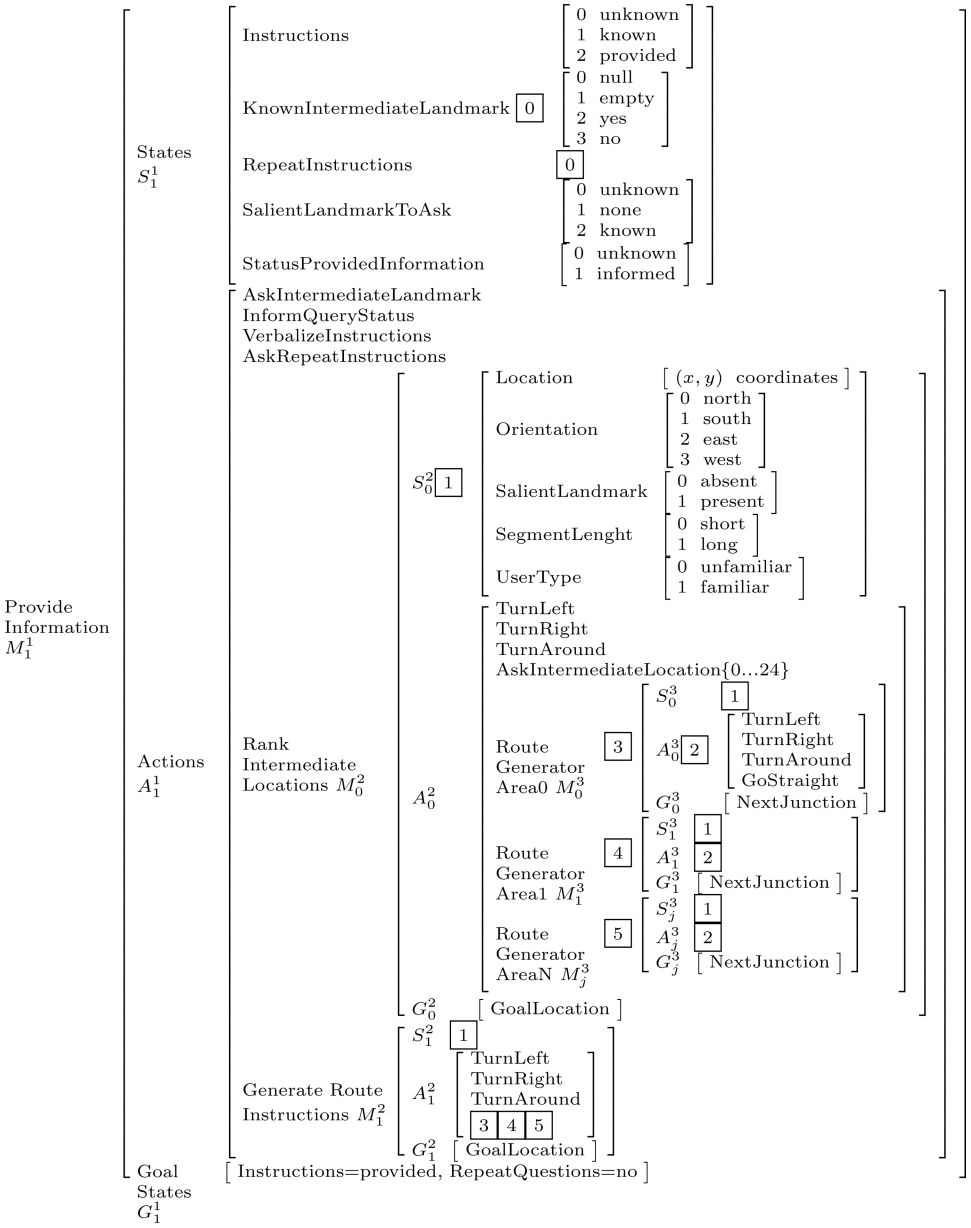

Fig. 2. Hierarchy of SMDP models (as feature structures) for the wayfinding dialogue system. The squared numbers show relationships in the hierarchy. The model M_1^1 unifies dialogue and spatial behavior, the models M_j^2 provide high-level navigation behavior by navigating to the goal by moving from one junction to another, and the models M_j^3 provide low-level navigation behavior by navigating with primitive actions. The models M_j^3 are illustrated in a compact way, they actually represent 82 models. After a user's query, the models with state variable "Location" involve dynamic goal states $\boxed{1} = \{(x, y), ?, ?, ?, ?)\}$, and they depend on the goal location (x, y).

where the variable "Confusion" has the value set {yes,no}. Because inducing such probabilities in a reliable way is beyond the scope of this work,[4] we used random probabilities in the range of $0 \leq 0.1$ for junctions with three or more segments and $0 \leq 0.05$ otherwise. Second, the user has beliefs about known locations in the environment, expressed by the conditional probability

$$P(Location|UserType). \tag{7}$$

For unfamiliar users we used a probability of 0.1 for every location in the environment, and for familiar users we used a probability of 0.8 for landmarks asked by the system (e.g., "Do you know how to get to the post room?"). By doing this, the learnt behavior for familiar users would tend to ask for intermediate locations, and for unfamiliar users would provide route instructions for the whole trajectory.

4.3. Experimental Results

Firstly, we observed that the learnt policies (using 1000 dialogues) generated behavior equivalent to the dialogues shown in Tables I and IV. Secondly, Figure 3 shows the learning curves of policies for all the test locations described in Figure 4, averaged over 10 training runs. It can be observed that our proposed learning approach learns much faster by reusing previously learned behavior. The fact that HSMQ-Learning with policy reuse stabilizes its behavior much more quickly than HSMQ-Learning without policy reuse, suggests that the former algorithm has promising application for online user-machine interaction. This confirms our previously formulated hypothesis that HSMQ-Learning with policy reuse is more suitable to optimize the behavior of spatially-aware dialogue systems. With regard to the set of learning curves in Figure 5, we can again observe the fast learning speed. We can also observe the significant quality improvements for policies that can exploit intermediate landmarks (tasks C, D, E) when the user is classified as familiar. This supports our remaining two hypotheses: a joint optimization of spatial and dialogue behaviors performs better than isolated optimization (since otherwise the agent would not have knowledge of suitable landmarks), and the agent learns to adapt its behavior to the prior knowledge of the user. A remaining question is how slow/fast our proposed learning approach would be perceived by real users. This question is addressed in the next section.

5. EXPERIMENTS USING A REALISTIC ENVIRONMENT

The main objective here was to show that our proposed approach, which is suitable for spatially-aware dialogue systems, can be used in real time with real users, that is, after a user's query, the learning agent induces a policy using the simulated environment in order to provide information to the user. For such a purpose we used a dialogue system for interactive indoor wayfinding. To the best of our knowledge this is the first reinforcement learning dialogue system using induced behavior in real-time.

5.1. Architecture of the Wayfinding Dialogue System

This dialogue system aims to provide users with route descriptions in English for navigating inside buildings that are generally recognized as presenting significant navigational challenges to both new and infrequent visitors. A pipeline architecture of this system is shown in the high-level diagram of Figure 6.[5] First, the user interacts

[4]We assume that it is not trivial to handcraft such beliefs for the whole spatial environment. In that case, the system needs feedback about the given route instructions in order to infer beliefs about confusing locations and known locations by the user. This is left as future work.

[5]A former version of our dialogue system using a simpler form of dialogue control is described in Cuayáhuitl et al. [2010b].

Table IV.
Sample dialogue showing the dynamics of policies of the right branch of the hierarchy. See Table I for the corresponding dialogue acts, here only primitive/composite actions are shown

Policy:State	Action (composite in italics)	Utterance (user response in italics)
π_0^0: ...	greeting	Welcome to the wayfinding info-kiosk.
π_0^0: ...	*collectInformation*	
π_0^1: ...	request	How can I help you? *How do I get to Dr. Watson's office?*
π_0^1: ...	clarify	Do you mean James or Peter Watson? *Peter Watson*
π_0^0: ...	*provideInformation*	
π_1^1:0, 0, 0, 0, 0	*rankIntermediateLocations*	
π_0^2:(2250, 1330), 2, 0, 0, 1	*routeGeneratorArea*$_j$	[high-level spatial behavior until goal]
π_j^3:(2250, 1330), 2, 0, 0, 1	turnLeft, goStraight, ...	[low-level spatial behavior until goal]
π_1^1:0, 0, 0, 2, 0	askIntermediateLocation	Do you know how to get to the stuga room? *No, I don't know that*
π_1^1:0, 3, 0, 2, 0	*generateInstructions*	
π_0^2:(2250, 1330), 2, 0, 0, 0	*routeGeneratorArea*$_j$	[high-level spatial behavior until goal]
π_j^3:(2250, 1330), 2, 0, 0, 0	turnAround, goStraight, ...	[low-level spatial behavior until goal]
π_1^1:0, 3, 0, 2, 0	informQueryStatus	OK, the following are your route instructions.
π_1^1:0, 3, 0, 2, 1	verbalizeInstructions	Turn around, then go straight on, passing the glass door, until the corridor. Turn left, and go straight on, passing the message board on your right until the stuga room. Turn to the right, then go straight, passing the glass door until the end of the corridor. Turn to the right, then go straight until the door A3445 on your right.
π_1^1:2, 3, 0, 2, 1	askRepeatInstructions	Do you want to hear the instructions again? *No thanks*
π_0^0: ...	askAnotherQuestion	Do you have another question? *No, good bye*
π_0^0: ...	closing	Thanks for using the wayfinding info-kiosk. Good bye!

with a Graphical User Interface (GUI) by asking questions about route directions using text-based natural language. Second, the language understanding module applies OpenCCG parsing [Clark et al. 2002] and keyword spotting—the latter is used in case of unparsed inputs—to the user utterance in order to extract a user dialogue act. Third, the dialogue management module specifies the system's behavior by mapping knowledge-compact dialogue states (extracted from the knowledge base that maintains the history of the interaction) to machine dialogue acts such as "request," "clarify," "presentInfo" or "askIntermediateLocation." Fourth, the language generation module provides high-level route instructions through the use of pCRU[6] that generates logical forms that are then realized by the KPML language surface generator [Bateman 1997], which in turn outputs text[7] in the GUI. Note that the spatially-aware dialogue manager interacts directly with the spatial data in order to perform a joint optimization

[6]pCRU is a probabilistic approach to resolving the nondeterminacy that typically arises in generation between a semantic representation and its possible linguistic surface forms, see [Belz 2008].
[7]We use template-based generation for all remaining system moves, e.g. opening, closing, etc.

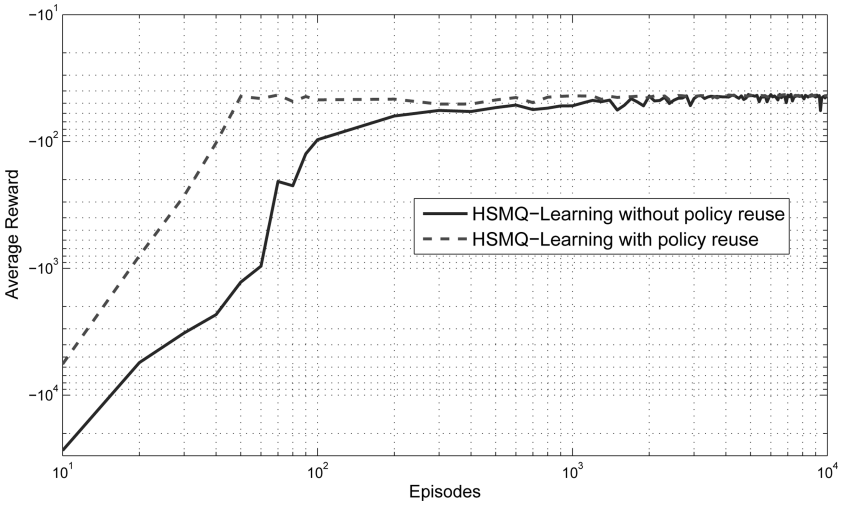

Fig. 3. Learning curves comparing the performance of two hierarchical reinforcement learning algorithms for spatially-aware dialogues in the wayfinding domain.

of dialogue and spatial behavior. For this purpose it uses the reinforcement learning approach described in Section 3.3.

We tested our wayfinding dialogue system in a university building that is typically considered complex to navigate. A map and a route graph of the environment are shown in Figure 4, where the white spaces represent (in our scenario nonnavigable) open spaces, a terrace and a courtyard. Although we only used data from a single floor, navigation on multiple levels using our proposed approach is still attractive for its fast learning (this may require an extra layer of models in the hierarchy).

5.2. Evaluation Methodology

We evaluated our dialogue system using objective and subjective metrics derived from the PARADISE framework [Walker et al. 2000]. This framework is commonly used for assessing the performance of spoken dialogue systems, and can be used for evaluating spatially-aware dialogue systems.

The following quantitative metrics were used for evaluation: dialogue efficiency, task success, and user satisfaction. First, the group of *dialogue efficiency* metrics includes "system turns," "user turns," and "elapsed time" (in seconds), the latter included the time it took the user to execute the task, that is, to find the given target destination. Second, the group of *task success* metrics includes the typical binary task success expressed as

$$\text{BinaryTaskSuccess} = \begin{cases} 1 & \text{for finding the target location} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

In addition, we used a graded task success metric which has shown higher correlation with user satisfaction than the previous metric [Dethlefs et al. 2010], expressed as

$$\text{GradedTaskSuccess} = \begin{cases} 1 & \text{for finding the target location} \\ 2/3 & \text{for finding the target location with small problems} \\ 1/3 & \text{for finding the target location with severe problems} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

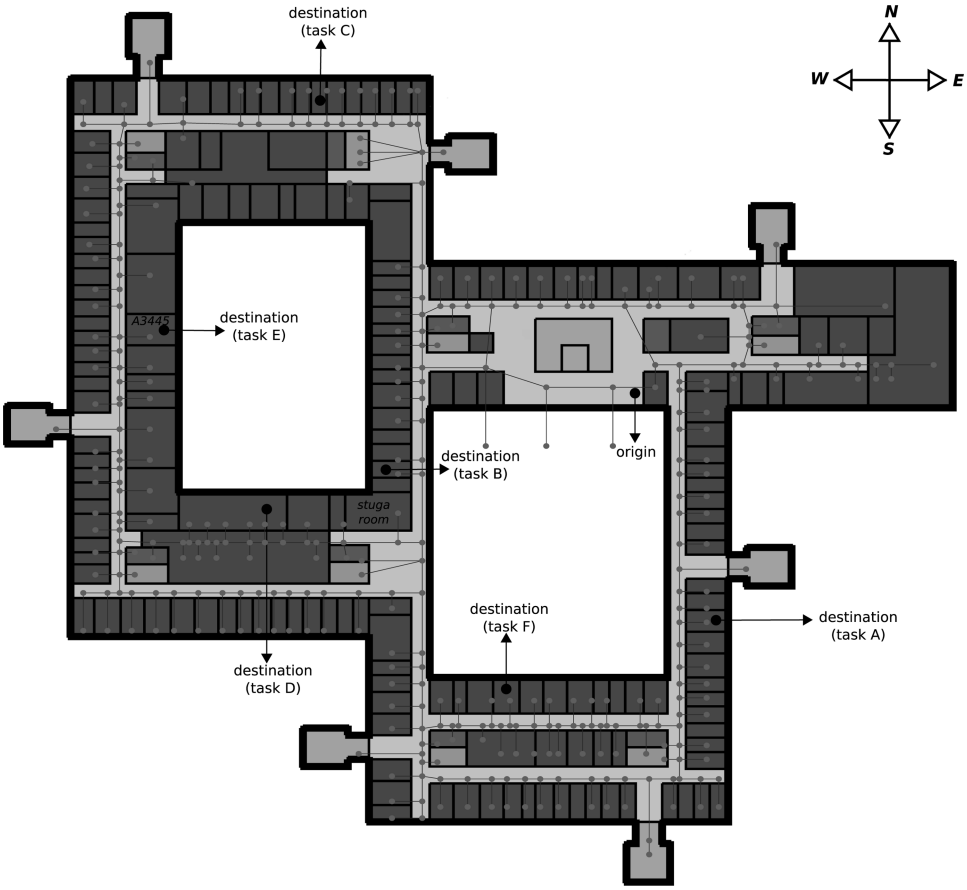

Fig. 4. Map of the navigation environment including a superimposed route graph, where the route graph specifies the navigational space and the white spaces represent open spaces, a terrace and a courtyard. The black circles represent origin and destination locations used in the evaluation (task E corresponds to the sample dialogues shown in Table I with initial orientation="east").

We assign the value of 1 when the user finds the target location without hesitation, the value with small problems when the user finds the location with slight confusion, and the value with severe problems when the user gets lost but eventually finds the target location. In both task success metrics, an experimental assistant, who followed users during the navigation tasks, assigned the values at the end of each task. In a correlation analysis between the assistant's task success ratings and execution time, we found a very high negative correlation (-0.96 at $p < 0.05$), which means that as the graded task success score decreases, the execution time increases. This is an indicator that the experimental assistant assigned reasonable scores. Finally, the group of *user satisfaction* metrics is described in Table V.

5.3. Experimental Setup

Twenty-two native speakers of German (with an average age of 23.3), who were university students enrolled in the "English-Speaking Cultures" course of studies, took part in our evaluation. Each user was presented with six wayfinding tasks, resulting in a total of 132 dialogues. They were asked in each case to find a particular location based

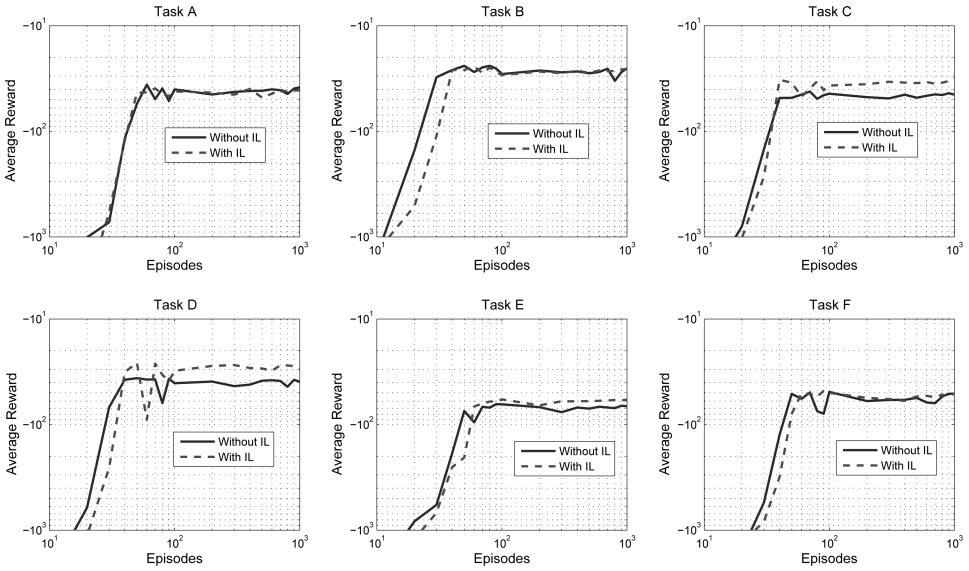

Fig. 5. Learning curves (using the HSMQ-Learning algorithm with policy reuse) comparing two types of behaviors for six navigation tasks: dialogue actions weakly coupled with spatial ones, and dialogue actions tightly coupled with spatial ones. The latter learns to ask for Intermediate Locations (IL), which is not trivial to specify. We can observe the quality improvements for policies that can exploit intermediate landmarks (tasks C, D, E). See Table I for sample dialogues, and Figure 4 for an illustration of the navigation tasks.

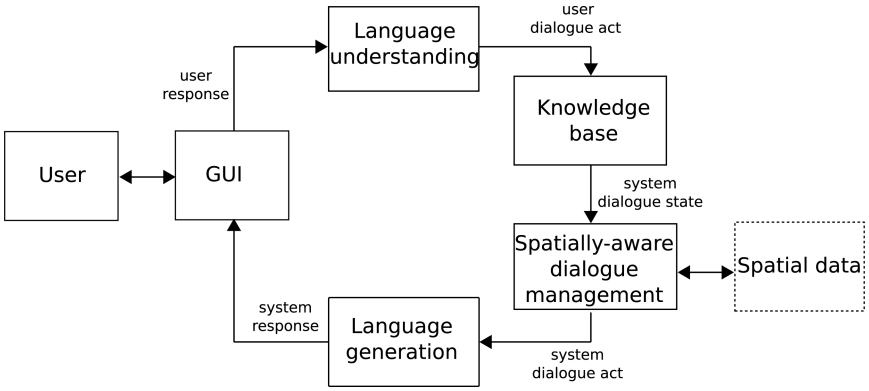

Fig. 6. A pipeline architecture of our reinforcement learning dialogue system for indoor navigation.

Table V. Subjective Measures for Evaluating Indoor Wayfinding

Measure	Question
Easy to Understand	Was the system easy to understand?
System Understood	Did the system understand what you asked?
Task Easy	Was it easy to find the location you wanted?
Interaction Pace	Was the pace of interaction with the system appropriate?
What to Say	Did you know what you could write at each point?
System Response	Was the system fast and quick to reply to you?
Expected Behavior	Did the system work the way you expected it to?
Future Use	Do you think you would use the system in the future?

Table VI.
Average results of our dialogue system (see sample dialogues in Table I). Note: we compared system performance for both user types. However, no significant differences were found except for system/user turns and "system words per instruction set," where familiar users received 18% fewer words than unfamiliar users (at $p < 0.01$) at the cost of one extra system-user turn

Group	Measure	Average Result
Dialogue Efficiency	System Turns	6.30
	User Turns	4.51
	System Words per Instruction Set	51.30
	User Words per Turn	3.79
	Time (minutes:seconds)	2:23
Task Success	Binary Task Success (%)	93.4
	Graded Task Success (%)	86.9
User Satisfaction	Easy to Understand	4.52
	System Understood	4.84
	Task Easy	4.24
	Interaction Pace	4.56
	What to Say	4.74
	System Response	4.48
	Expected Behavior	4.40
	Future Use	3.82
	Sum	35.6/40

on the route instructions generated by the dialogue system on request of the user. The system was located at a static point (marked "origin" on the map provided in Figure 4), where users would return after each wayfinding task to request the next location. The locations were spatially distributed (see Figure 4). The dialogue tasks were executed pseudorandomly (from a uniform distribution), in order to prevent ordering effects on the ratings of participants. In addition, we alternated the user type (familiar or unfamiliar), in the former the learning agent tended to ask for intermediate locations. At the beginning of each session, participants were asked about their familiarity with the building using a 5-point Likert scale, where 1 represents the lowest familiarity and 5 the highest. This resulted in an average familiarity score of 2.7. Then, our participants received the following set of instructions: (a) you can ask the system using natural language, (b) you can take notes from the received instructions, (c) follow the instructions as precisely as possible, (d) you are not allowed to ask anyone how to get to the target location, and (e) you can give up anytime after trying without success by telling that to the assistant that will follow you. At the end of each wayfinding task, participants were asked to fill a questionnaire (Table V) for obtaining qualitative results using a 5-point Likert scale, where 5 represents the highest score.

5.4. Experimental Results

Evaluation results of the system described in this section are summarized in Table VI, where two main results can be drawn from this evaluation. First, reinforcement learning in real time—using our proposed approach—is feasible in terms of execution time.[8] It can be observed that users rated the speed of system response with a score of 4.48 out of 5. Second, the spatially-aware dialogue behavior was considered reasonable by the users. This can be noted from the metrics "easy to understand" and "task easy" with scores of 4.52 and 4.24, respectively. It can also be observed that the lowest qualitative score was given to the metric "future use" (3.82), presumably for the following reasons:

[8]Our learning dialogue system ran on a machine with minimal contemporary requirements. Specifically, using a laptop with a core 2 duo processor and 2 GB of RAM, running Windows professional.

(a) users may prefer maps over text; and (b) new or infrequent visitors to the building may find the system more useful than users with partial knowledge of the building.

In addition, we compared system performance for both types of users. However, no significant differences were found except for dialogue turns and "system words per instruction set," where familiar users received 18% less words than unfamiliar users (at $p < 0.01$) at the cost of one extra system-user turn. Presumably, even when familiar users received more compact instructions, they also had to navigate to intermediate locations, resulting in task duration times that are equivalent to those of unfamiliar users. Future work can evaluate in-situ dialogues and induce the user type/confusions on the fly so that the learning agent can adapt accordingly.

Finally, we included an additional question in the survey filled after each dialogue: "Did you find the location exclusively based on the instructions given by the system or did you use additional help such as signs?" This question also used a 5-point Likert scale, where 5 represents the highest score for strictly following only the system instructions. This resulted in an average value of 4.2, suggesting that the results we have given were derived from following almost entirely the system's instructions.

6. RELATED WORK AND DISCUSSION

The interdisciplinary topic of this article addresses the fields of machine learning, dialogue systems, and spatial cognition. We discuss the following subjects to position our article in the literature: (1) role of hierarchical control; (2) automatic route instruction generation; (3) reinforcement learning dialogue systems; (4) spatially-aware dialogue control using reinforcement learning; (5) reinforcement learning in real time, during the course of the interaction; (6) joint optimization with other system modules; and (7) simulated behavior for spatially-aware dialogue behaviors.

First, the importance of hierarchical learning is to perform a more scalable optimization. This form of learning is also important to optimize decision making at different levels of granularity, where the design of the subtask sequence might not be easy to handcraft. For instance, in the wayfinding domain: When to ask for intermediate locations (which aim to provide compact route instructions)? What route to follow? This scenario requires learning at low and high levels in the hierarchy to result in a unified dialogue policy. Moreover, the importance of hierarchical learning increases according to the complexity and size of the state-action space of a given system. This makes the optimization of applications with large and complex behavior feasible, where the division of a behavior into subbehaviors produces faster learning, reduces computational demands, and provides opportunities to reuse subsolutions. The latter proved essential for fast learning, which provides us with a mechanism for learning behavior in real time. However, little attention has been given to *hierarchical learning*, which has been mostly applied to small-scale dialogue systems using policies with frozen learning in real time [Pineau 2004; Lemon 2010; Rieser and Lemon 2008; Janarthanam and Lemon 2010].

Second, adaptive route instruction generation in the spatial cognition community has typically been oriented on the principles that humans employ in wayfinding, thus making instructions cognitively adequate from a human perspective. Such principles include the use of salient landmarks at decision points or along the route, spatial chunking, or tailoring of instructions towards the spatial environment or different user groups. Several approaches have taken such information into account for route generation. Richter and Duckham [2008] generated routes using the path that is simplest to describe for the system. In contrast, Duckham and Kulik [2003] generated routes using the path that is easiest for users to follow. While such approaches achieve alignment with human route descriptions, they only adapt to a majority of users. They fail, however, to adapt to the information needs of specific individuals. For example, users

familiar with the navigation environment might prefer a short, schematic description, while unfamiliar users require more guidance. Similarly, familiar users might prefer the shortest route to the destination, whereas unfamiliar users might prefer the easiest. There is no single route description that satisfies both users' needs simultaneously. Cuayáhuitl et al. [2010a] addressed the scenario of generating different types of routes according to users' familiarity with the navigation environment, but they did not take into account dialogic interaction.

All of these systems include features of adaptation or cognitive and linguistic adequacy that the present work will also need to address. However, they mostly focus on route instructions in outdoor environments. Related work in indoor navigation has so far put a strong emphasis on generating visual support for users, rather than optimizing the content and form of route instructions or their textual choices. Kray et al. [2005] presented an interactive display system that is mounted on walls and provides visual navigation support to building users. Callaway [2007] presented a system that assists users while navigating through an indoor environment, rather than providing in-advance instructions. Münzer and Stahl [2007] describe modeling software that generates dynamic visual route information, and Hochmair [2008] reported on a desktop usability study comparing various models of indoor navigation aids. Further, Becker et al. [2008] and Ohlbach and Stoffel [2008] presented models for representing complex spatial configurations adequately for navigation and route assistance. Kruijff et al. [2007] presented a human-robot interaction scenario set within an office environment. All of these approaches have put a primary focus on providing users with visual support for their navigation environment. Further, they have tended to take the cognitive and linguistic quality of their instructions for granted and do not include any of the adaptation effects addressed briefly above and laid out in more detail in Section 2. Cuayáhuitl et al. [2010b] presented a system for in-advance, text-based route instructions that uses landmarks and spatial chunking, but they leave unaddressed the adaptation to users' information needs or prior knowledge.

Third, while we are not aware of any prior work that applied reinforcement learning to interactive wayfinding, there are several approaches in the dialogue community that have applied it to information-seeking dialogue systems, or for the optimization of decisions of content selection or information presentation. In the area of information-seeking dialogue systems, Levin et al. [2000] presented a system that uses RL to learn dialogue strategies in an air travel information system. Similarly, Walker [2000] learned dialogue strategies for a spoken dialogue system that lets users access their email over telephone, and Singh et al. [2002] optimized dialogue policies in a system that provides information to tourists. Cuayáhuitl et al. [2010c] applied hierarchical reinforcement learning to dialogue strategy learning in a spoken dialogue system operating in the travel planning domain. While all of these systems use RL to optimize aspects of system behavior, none of them is a situated dialogue system. Therefore, they do not require knowledge that is peculiar to their domain of application, and they can act blindly with respect to the current pragmatic situation of the dialogue. This is in contrast to our own work.

Fourth, reinforcement learning in the language processing community has recently been adopted for building adaptive human-machine conversational systems. While most of the attention has been given to information-seeking dialogue systems, spatially-aware dialogue systems have received little attention. For example, work at Karlsruhe University has investigated human-robot interaction using reinforcement learning [Stiefelhagen et al. 2007]; however, they do not provide a unified optimization between the dialogue and spatial behaviors. Our work addresses this latter issue, which was crucial for our domain in order to provide adaptive spatially-aware behavior (e.g., by inducing wayfinding behaviors such as "Do you know how to get to the lifts?"),

and can be applied to other dialogue systems requiring spatial awareness. We are not aware of any other reinforcement learning dialogue system in the situated domain. Other situated dialogue systems in the literature do not optimize dialogue behavior and treat it independently of the spatial behavior. Examples of such kind of systems are the WITAS dialogue system [Lemon et al. 2001], the DFKI's human-robot interaction dialogue system [Kruijff et al. 2007], or the Daisie framework for situated interaction [Ross and Bateman 2009].

Fifth, our work differs from the work at Karlsruhe University by the use of dynamic goal states, which require policy learning in *real-time* (after a user's query such as "How do I get to room B3044?"). If the dialogue system is aiming to provide adaptive behavior to the user and environmental space, then the dialogue manager must be tightly integrated with spatial behaviors. Our learning dialogue system is innovative in policy learning in real time (using simulations). This is required for navigating from any point in the space to any other, and to adapt to a dynamic environment. In addition, in order to avoid learning from scratch, policy reuse has been addressed in the literature, but only for flat reinforcement learning [Lemon et al. 2006; Fernández and Veloso 2006]. Our learning approach is based on policy reuse for fast learning. We are not aware of any other prior work with policy reuse in a hierarchical setting, and much more remains to be done in this direction.

Sixth, it has been shown before that *joint optimizations* are better than independent ones. For example, Lemon [2010] optimizes dialogue strategies with content selection strategies in a tourist information system. In addition, Janarthanam and Lemon [2010] optimize a dialogue-generation policy that adapts its behavior to expert and novice users with varying degree of technical terms. Our work also exhibits a joint optimization between dialogue management and route instruction generation, where our proposed learning approach provides the mechanisms for doing such joint optimizations in a scalable way. We argue that the hierarchical setting plays an important role for such a purpose in order to optimize more complex behaviors. For example, our learning dialogue agent can jointly optimize confirmation strategies for spoken interaction [Cuayáhuitl et al. 2010c], natural language generation strategies for adaptive text generation [Dethlefs and Cuayáhuitl 2010], and multimodal behavior [Prommer et al. 2006; Wyatt 2005], among others.

Finally, a typical approach to model simulated user behavior is by estimating probabilistic models (e.g., n-gram models) from data [Schatzmann et al. 2006]. However, collecting data to estimate such models for spatially-aware dialogue behavior is more difficult because of the vast amount of required data. In the particular case of wayfinding, a potential approach is to estimate such conditional probabilities from in-situ dialogues with real users, where the system would be aware of user confusions given some spatial features and therefore keep an up-to-date model of the environment. By estimating a more reliable model of the spatial environment, the learning agent can adapt its behavior accordingly, and therefore make the reinforcement learning approach more attractive for situated dialogue systems.

7. CONCLUSION AND FUTURE WORK

In this article we have described a hierarchical reinforcement learning approach for optimizing the behavior of spatially-aware conversational interfaces, which performs learning in advance (using a static simulation environment) and learning in real-time (using a dynamic simulation environment). For such a purpose we extended an existing algorithm in the literature of reinforcement learning in order to support reusable policies and therefore to perform fast policy optimization in real-time settings. We evaluated our approach by incorporating it into a text-based dialogue system for indoor navigation. The novelty in our dialogue system is the joint optimization in real

time between dialogue and spatial behaviors, suitable for dynamic environments. Our experimental results provide evidence to conclude that our reinforcement learning approach is promising because it combines fast learning with adaptive and reasonable behavior. This claim is supported by a quantitative evaluation reporting high binary task success (93%) and a qualitative evaluation reporting high user satisfaction (89%).

This research makes the following contributions to situated dialogue systems that learn their dialogue behavior. The first contribution is a reinforcement learning approach that jointly optimizes dialogue and spatial behaviors in real time. This approach is based on two fundamental concepts: hierarchical control and policy reuse. While hierarchical control is used to simplify the problem by dividing a learning task into subtasks, policy reuse is the mechanism used to accelerate learning. The second contribution is the application of our proposed approach to a dialogue system in the wayfinding domain. Our experiments show that this approach can be used to optimize dialogue actions tightly coupled with spatial ones, resulting in a joint optimization (which we call spatially-aware dialogue control). In addition, our proposed approach was fast enough to guide users in navigation tasks without significant time delays, according to our evaluation with real users. In summary, our unified and scalable learning approach provides a framework that can be used to optimize more complex behavior in different domains.

The work we have described opens an exciting direction for spatially-aware dialogue systems using hierarchical reinforcement learning. We suggest the following research avenues to make progress in this field:

—To induce reliable simulated environments for learning adaptive spatially-aware dialogue behavior. A suitable computational approach to use is "Bayesian networks" in order to integrate beliefs about dialogue and spatial knowledge.
—To evaluate our proposed reinforcement learning approach in other scenarios for showing adaptive behavior to a changing spatial environment and to user's prior knowledge, where objects constantly change their level of salience. An example could be in-situ dialogues (i.e., the system interacts with the user as he/she carries out the navigation task) using a hand-held device and spoken interaction.
—To carry out joint optimizations with other behaviors such as confirmation strategies, natural language generation and multimodal interaction.
—To constrain the spatially-aware dialogue behavior with some form of prior knowledge and other forms of policy reuse, allowing further speed-up in learning. A potential approach to follow is hierarchical relational reinforcement learning.
—To investigate effective reward functions (induced from data) for spatially-aware dialogue systems using hierarchical reinforcement learning. A potential approach to follow is hierarchical model-based reinforcement learning. This is relevant since large-scale joint optimizations make the reward function more complex to specify.
—To extend our proposed reinforcement learning approach with other approaches such as function approximation and belief monitoring. While the former enhances scalability, the latter is useful to support interactions under uncertainty.

ACKNOWLEDGMENTS

We thank Lutz Frommberger, Kai-Florian Richter, John Bateman, Thora Tenbrink, Elena Andonova, Cui Jian and Michael TRB Turnbull for helpful discussions. We also thank our anonymous reviewers whose comments improved the clarity of this article.

REFERENCES

BATEMAN, J. A. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. *J. Nat. Lang. Engin. 3,* 1, 15–55.

BECKER, T., NAGEL, C., AND KOLBE, T. H. 2008. A multilayered space-event model for navigation in indoor spaces. In *Proceedings of the 3rd International Workshop on 3D Geo-Information*. J. Lee and S. Zlatanova Eds., Springer, Berlin.

BELZ, A. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Nat. Lang. Engin. 1*, 1–26.

BURNETT, G., SMITH, D., AND MAY, A. 2001. Supporting the Navigation Task: Characteristics of 'G'ood Landmarks. *Contemp. Ergonom.* 441–446.

CALLAWAY, C. 2007. Non-localized, interactive multimodal direction giving. In *Proceedings of the Workshop on Multimodal Output Generation (MOG'07)*. I. van der Sluis, M. Theune, E. Reiter, and E. Krahmer Eds., Centre for Telematics and Information Technology (CTIT), University of Twente, 41–50.

CLARK, S., HOCKENMAIER, J., AND STEEDMAN, M. 2002. Building deep dependency structures using a wide-coverage CCG parser. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. 327–334.

CUAYÁHUITL, H. 2009. Hierarchical reinforcement learning for spoken dialogue systems. Ph.D. thesis, School of Informatics, University of Edinburgh.

CUAYÁHUITL, H., DETHLEFS, N., FROMMBERGER, L., RICHTER, K.-F., AND BATEMAN, J. 2010a. Generating adaptive route instructions using hierarchical reinforcement learning. In *Proceedings of the International Conference on Spatial Cognition (Spatial Cognition VII)*.

CUAYÁHUITL, H., DETHLEFS, N., RICHTER, K.-F., TENBRINK, T., AND BATEMAN, J. 2010b. A dialogue system for indoor wayfinding using text-based natural language. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*.

CUAYÁHUITL, H., RENALS, S., LEMON, O., AND SHIMODAIRA, H. 2007. Hierarchical dialogue optimization using Semi-Markov decision processes. In *Proceedings of INTERSPEECH*. 2693–2696.

CUAYÁHUITL, H., RENALS, S., LEMON, O., AND SHIMODAIRA, H. 2010c. Evaluation of a hierarchical reinforcement learning spoken dialogue system. *Comput. Speech Lang. 24,* 2, 395–429.

DENECKE, M., DOHSAKA, K., AND NAKANO, M. 2004. Fast reinforcement learning of dialogue policies using stable function approximation. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*. 1–11.

DENIS, M. 1997. The description of routes: A cognitive approach to the production of spatial discourse. *Cahiers Psychologie Cognitive 16,* 4, 409–458.

DETHLEFS, N. AND CUAYÁHUITL, H. 2010. Hierarchical reinforcement learning for adaptive text generation. In *Proceedings of the International Conference on Natural Language Generation (INLG)*.

DETHLEFS, N., CUAYÁHUITL, H., RICHTER, K.-F., ANDONOVA, E., AND BATEMAN, J. 2010. Evaluating task success in a dialogue system for indoor navigation. In *Proceedings of the 14th Workshop on the Semantics and Pragmatics of Dialogue (SemDial)*.

DIETTERICH, T. 2000a. Hierarchical reinforcement learning with the MAXQ value function decomposition. *J. Artif. Intell. Resear. 13,* 1, 227–303.

DIETTERICH, T. 2000b. An overview of MAXQ hierarchical reinforcement learning. In *Proceedings of the Symposium on Abstraction, Reformulation, and Approximation*. 26–44.

DUCKHAM, M. AND KULIK, L. 2003. "Simplest" paths: Automated route selection for navigation. In *Spatial Information Theory*, W. Kuhn, M. Worboys, and S. Timpf Eds., Lecture Notes in Computer Science, vol. 2825, Springer, Berlin, 169–185.

FERNÁNDEZ, F. AND VELOSO, M. 2006. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 720–727.

HENDERSON, J., LEMON, O., AND GEORGILA, K. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computat. Ling. 34,* 4, 487–511.

HOCHMAIR, H. H. 2008. PDA-assisted indoor-navigation with imprecise positioning: Results of a desktop usability study. In *Map-Based Mobile Services: Interactivity, Usability and Case Studies*, L. Meng, A. Zipf, and S. Winter Eds., Springer, Berlin, 228–247.

JANARTHANAM, S. AND LEMON, O. 2010. Learning to adapt to unknown users: referring expression generation in spoken dialogue systems. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. 69–78.

KAELBLING, L., LITTMAN, M., AND MOORE, A. 1996. Reinforcement learning: A survey. *J. Artifi. Intell. Resear. 4*, 237–285.

KLIPPEL, A., HANSEN, S., RICHTER, K.-F., AND WINTER, S. 2009. Urban granularities - a data structure for cognitively ergonomic route directions. *GeoInformatica 13,* 2, 223–247.

KLIPPEL, A., TENBRINK, T., AND MONTELLO, D. R. 2010. The role of structure and function in the conceptualization of directions. In *Motion Encoding in Language and Space*, E. van der Zee and M. Vulchanova Eds., Oxford University Press.

KRAY, C., KORTUEM, G., AND KRÜGER, A. 2005. Adaptive navigation support with public displays. In *Proceedings of Conference on Intelligent User Interfaces (IUI). ACM*, R. S. Amant, J. Riedl, and A. Jameson Eds., ACM, NY, 326–328.

KRUIJFF, G., ZENDER, H., JENSFELT, P., AND CHRISTENSEN, H. 2007. Situated dialogue and spatial organization: What, where... and why? *Int. J. Adv. Robo. Syst. 4,* 2. (Special Issue on Human and Robot Interactive Communication.)

LEMON, O. 2010. Learning what to say and how to say it: Joint optimization of spoken dialogue management and natural language generation. *Comput. Speech Lang.*

LEMON, O., BRACY, A., GRUENSTEIN, A., AND PETERS, S. 2001. The WITAS multi-modal dialogue system I. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*. 1559–1562.

LEMON, O., GEORGILA, K., AND HENDERSON, J. 2006. Evaluating efectiveness and portability of reinforcement learned dialogue strategies with real users: The TALK TownInfo evaluation. In *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT)*. 178–181.

LEVIN, E., PIERACCINI, R., AND ECKERT, W. 2000. A stochastic model of human machine interaction for learning dialog strategies. *IEEE Trans. Speech Audio Proc. 8,* 1, 11–23.

LOVELACE, K. L., HEGARTY, M., AND MONTELLO, D. R. 1999. Elements of good route directions in familiar and unfamiliar environments. In *Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science*, Lecture Notes in Computer Science, vol. 1661.

MAY, A. J., ROSS, T., AND BAYER, S. H. 2003. Drivers' Information Requirements when Navigating in an Urban Environment. *J. Navigation 56,* 01, 89–100.

MÜNZER, S. AND STAHL, C. 2007. Providing individual route instructions for indoor wayfinding in complex, multi-level buildings. In *Proceedings of the 5th Geographic Information Days*. F. Probst and C. Keßler Eds., 241–246.

OHLBACH, H. J. AND STOFFEL, E.-P. 2008. Versatile route descriptions for pedestrian guidance in buildings: Conceptual model and systematic method. In *Proceedings of the 11th AGILE International Conference on Geographic Information Science*.

PIETQUIN, O. 2004. A framework for unsupervised learning of dialogue strategies. Ph.D. thesis, Faculté Polytechnique de Mons.

PINEAU, J. 2004. Tractable planning under uncertainty: Exploiting structure. Ph.D. thesis, Carnegie Mellon University.

PROMMER, T., HOLZAPFEL, H., AND WAIBEL, A. 2006. Rapid simulation-driven reinforcement learning of multimodal dialog strategies in human-robot interaction. In *Proceedings of the INTERSPEECH*. 1918–1921.

RAUBAL, M. AND WINTER, S. 2002. Enriching wayfinding instructions with local landmarks. In *Proceedings of the 2nd International Conference on Geographic Information Science, 2002*, M. Egenhofer and D. Mark Eds., Lecture Notes in Computer Science, vol. 2478, Springer, Berlin, 243–259.

RICHTER, K.-F. AND DUCKHAM, M. 2008. Simplest instructions: Finding easy-to-describe routes for navigation. In *Proceedings of the 5th International Conference on Geographic Information Science*, T. J. Cova, H. J. Miller, K. Beard, A. U. Frank, and M. F. Goodchild Eds., Lecture Notes in Computer Science, vol. 5266. Springer, Berlin, 274–289.

RIESER, V. AND LEMON, O. 2008. Learning effective multimodal dialogue strategies from Wizard-of-Oz data: Bootstrapping and evaluation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. 638–646.

ROSS, R. J. AND BATEMAN, J. A. 2009. Daisie: Information state dialogues for situated systems. In *Proceedings of the International Conference on Text, Speech and Dialogue*. Lecture Notes in Compter Science, vol. 5729. Springer, 379–386.

RUSSELL, S. AND NORVIG, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education.

SCHATZMANN, J., WEILHAMMER, K., STUTTLE, M., AND YOUNG, S. 2006. A survey on statistical user simulation techniques for reinforcement learning of dialogue management strategies. *Knowl. Engin. Rev. 21,* 2, 97–126.

SCHEFFLER, K. 2002. Automatic design of spoken dialogue systems. Ph.D. thesis, Cambridge University.

SINGH, S., LITMAN, D., KEARNS, M., AND WALKER, M. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *J. Artif. Intell. Resear. 16*, 105–133.

Sorrows, M. E. and Hirtle, S. C. 1999. The nature of landmarks for real and electronic spaces. In *Spatial Information Theory*, C. Freksa and D. M. Mark Eds., Lecture Notes in Compter Science, vol. 1661, Springer, 37–50. LNCS 1661.

Stiefelhagen, R., Ekenel, H., Fugen, C., Gieselmann, P., Holzapfel, H., Kraft, F., Nickel, K., Voit, M., and Waibel, A. 2007. Enabling multimodal human-robot interaction for the Karlsruhe humanoid robot. *IEEE Trans. Robot. 23,* 5, 840–851.

Sutton, R. and Barto, A. 1998. *Reinforcement Learing: An Introduction*. MIT Press.

Tenbrink, T. and Winter, S. 2009. Variable granularity in route directions. *Spat. Cognit. Computat. 9,* 1, 64–93.

Thomson, B. 2009. Statistical methods for spoken dialogue management. Ph.D. thesis, University of Cambridge.

Toney, D. 2007. Evolutionary reinforcement learning of spoken dialogue strategies. Ph.D. thesis, University of Edinburgh.

Walker, M. 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *J. Artif. Intell. Resear. 12*, 387–416.

Walker, M., Kamm, C., and Litman, D. 2000. Towards developing general models of usability with PARADISE. *Nat. Lang. Engine. 6,* 3, 363–377.

Werner, S., Krieg-Brâckner, B., and Herrmann, T. 2000. Modelling navigational knowledge by route graphs. In *Spatial Cognition II*, E. A. Freksa Ed., Lecture Notes in Compter Science, vol. 1849, Springer, 295–316.

Williams, J. 2006. Partially observable Markov decision processes for spoken dialogue management. Ph.D. thesis, Cambridge University.

Wyatt, J. 2005. Planning clarification questions to resolve ambiguous references to objects. In *Proceedings of the Workshop on Knowledge and Reasoning in Practical Dialogue Systems (IJCAI)*.

Young, S. 2000. Probabilistic methods in spoken dialogue systems. *Phil. Trans. Roy. Soc. (Series A) 358,* 1769, 1389–1402.

Young, Y., Gasic, M., Keizer, S., Mairesse, F., Schatzmann, J., B., T., and Yu, K. 2010. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. *Comput. Speech Lang. 24,* 2, 150–174.

Received July 2010; revised November 2010; accepted December 2010