A Decision Theoretic Approach for Task Coordination in Social Robots

Pantelis Elinas
Dept. of Computer Science
Univ of British Columbia
2366 Main Mall
Vancouver, BC,
Canada, V6T 1Z4

Enrique Sucar and Alberto Reyes
Dept. of Computer Science
ITESM Cuernavaca
Reforma #182-A
Temixco, Morelos
62589 Mexico

Jesse Hoey
Dept. of Computer Science
University of Toronto
6 King's College Road
Toronto, Ontario
Canada M5S 3H5

Abstract

We present a new method for designing and implementing socially interactive mobile robots built around a 3-layer hybrid control architecture. Our main contribution is at the deliberative level, where we introduce Multiply Sectioned Markov Decision Processes (MS-MDPs) as a mechanism for efficient task specification, policy generation and execution. Using MS-MDPs, we partition the task into a number of subtasks, each assigned to an MDP, such that each one can be specified and solved independently. Each MDP controls one aspect of the global task, and they all are executed concurrently, coordinated implicitly by common state variables. We validate our approach by presenting experiments performed using our robot HOMER, the Human Oriented MEssenger Robot. HOMER is a stereo-vision guided mobile robot designed for performing a message delivery task, which allows for rich and complex robot-human interactions using a multi-modal interface. HOMER's deliberative layer includes 3 MDPs: the navigator, the dialogue manager and the gesture generator. Together they coordinate 10 behaviors for accomplishing the message delivery task.

1 Introduction

We are concerned with the problem of building mobile robotic systems with capacities to interact with humans. Such robots will need navigation, mapping, localization and obstacle avoidance capabilities to deal with moving around in an uncertain and changing environment. They will also need to model the dynamics of people in an environment, including their locations in space and their behavioral patterns. Finally, human users will require such robots to present clear, simple and natural interactive interfaces, which enable easy exchanges of information between robot and human. The systems to deal with each of these problems can be made fairly independent, and thus can be implemented modularly. The remaining challenge is to integrate and coordinate the modules to perform a given task.

We propose a framework for task coordination based on multiple Markov decision processes (MDPs) that satisfies the previous requirements, which we call *Multiply Sectioned Markov Decision Processes (MS-MDPs)*. Using a representation based on MS-MDPs, we partition the task into a number of subtasks, each assigned to an MDP, such that each one can be specified and solved independently. Each MDP controls one aspect of the global task, and all are executed concurrently, coordinated implicitly by common state variables. At the execution stage, all MDP policies are consulted concurrently, and the best actions (for each) are selected according to each policy and the current state.

The MS-MDP approach has several advantages against using a single MDP: (i) it is easier to specify a simpler, smaller MDP for each subtask, (ii) it is more efficient in terms of solving each subtask MDP versus solving a more complex single MDP. However, these efficiency gains come at the cost of optimality, as we discuss in Section 4.2. MS-MDPs also have an advantage over hierarchical MDPs [4], since they allow multiple actions to be executed simultaneously without considering all possible action combinations.

We validate our approach by presenting experiments performed using our robot: HOMER. HOMER, the Human Oriented MEssenger Robot, is a mobile robot that communicates messages between humans in a workspace. The message delivery task is a challenging domain for an interactive robot. It presents all the difficulties associated with uncertain navigation in a changing environment, as well as those associated with exchanging information and taking commands from humans using a natural interface. For this task we use 3 MDPs: the navigator, the dialogue manager and the gesture generator. Together they coordinate 10 behaviors for accomplishing the message delivery task. Although we describe our framework for the message delivery task, our system can easily be extended to other human-interactive mobile robotic tasks.

The rest of the paper is organized as follows. We begin by reviewing related work in social mobile robots, in particular in alternative approaches for planning and coordination. Then we introduce our mobile robot, HOMER, describe his hardware and software systems, and the different modules that are used for the message delivery task. We describe the general framework for task coordination based on MS-MDPs, and present the particular configuration used for HOMER. We then introduce the domain of message delivery, and show results of some experiments demonstrating HOMER's performance in a complex domain. Lastly, we conclude and discuss future research directions.

2 Related work

Building service robots to help people has been the subject of much recent research. The challenge is to achieve reliable systems that operate in highly dynamic environments and have easy to use interfaces. This involves solving both the more traditional robot problems of navigation and localization and the more recent problems in human-robot interaction. Another challenge arises from the large scope of these systems and the many pieces that must be integrated together to make them work. RHINO [3], was one of the most successful service robots ever built. It was designed as a museum tour guide. RHINO successfully navigated a very dynamic environment using laser sensors and interacted with people using pre-recorded information; a person could select a specific tour of the museum by pressing one of many buttons on the robot. RHINO's task planning was specified using an extension to the GOLOG language called GOLEX; GOLEX is an extension of first order calculus, but with the added ability to generate hierarchical plans and a run-time component monitoring the execution of those plans. MINERVA [17], was the successor of RHINO. MINERVA differed from RHINO in that it could generate tours of exhibits in real-time as opposed to choosing one of several pre-determined tours. MINERVA also improved on the interaction by incorporating a steerable head capable of displaying different emotional states. The GOLOG language was combined with decision theoretic planners in DTGOLOG, used in the implementation of a service delivery robot [2]. More recently, the robot PEARL escorted elderly people around an assisted living facility [10]. Its navigation and localization used probabilistic techniques with laser sensors. PEARL is more focused on the interaction side with an expressive face and a speech recognition engine.

One of PEARL's contributions is the use of a hierarchical partially observable Markov decision process (HPOMDP), which is an extension of hierarchical MDPs (HMDPs) [4] to model uncertain observations. HMDPs use an specified hierarchical breakdown of the domain, and introduce *abstract actions* in higher level MDPs which invoke the policies of lower-level MDPs. Our framework of Multiply-Sectioned MDPs (MS-MDPs) is related to HMDPs, with the important difference that MS-MDPs do not require recursive execution of policies, instead allowing multiple actions to be performed simultaneously. When a subtask is given control in a HMDP, no other subtask can interrupt it until it completes and returns control to the level in the hierarchy that invoked it. In a MS-MDP, on the other hand, subtasks are all executed concurrently, allowing multiple behaviors in a robotic system to be controlling the actions of the robot simultaneously. Markov Task Sets [9] also allow this type of concurrent control, but assume utility independence between subtasks. MS-MDPs use a common reward function for all subtasks, and seek collaborative solutions between behaviors.

The work we present in this paper builds upon our previous service robots, including the award-winning waiter, José [5], and an initial implementation of HOMER [6], in which we used a single Markov decision process (MDP) planner. Our current work extends this by adding additional behaviors, and by introducing MS-MDPs to make the planning more efficient.

(a) (b)

Figure 1: (a) HOMER the messenger robot interacting with a person and (b) closeup of HOMER's head

3 HOMER

3.1 Hardware

Our robot, HOMER, shown in Figure 1(a), is a Real World Interface B-14 robot, and has a single sensor: a Point Grey Research Bumblebee^{TM} stereo vision camera. The Bumblebee is mounted atop an LCD screen upon which is displayed a pleasant and dynamic animated face. We call the combination of Bumblebee and LCD screen *the head*. The face displays non-verbal invitations to humans to approach and speak, expresses emotions, or emphasizes or conveys further information. The head is mounted on a Directed Perception pan-tilt unit, as shown in Figure 1(b), which provides lateral and dorsal movement for the camera system and animated face, enabling visual search and following for realistic interaction. An omnidirectional microphone is the

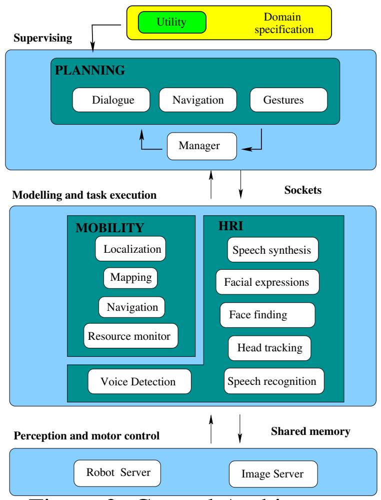

Figure 2: Control Architecture

only other sensor available on HOMER and it is used for speech recognition.

HOMER is equipped with 4 on board computers based on the Intel Pentium III processor and running the LINUX operating system. The computers communicate among each other using a 100Mbps local area network. A Compaq wireless 10Mbps network connection allows the robot's computer to communicate with any other computer in our lab's LAN for additional compute power as needed.

3.2 Software Architecture

Crucial to the design of a human-interactive mobile robot is the ability to rapidly and easily modify the robot's behavior, specifying the changes in a clear and simple language of high-level concepts. These design constraints call for a mobile robot to have a modular software system, with a planning module written in a simple and easy to use language for specifying new world states and goals. HOMER's software system is designed around a Behavior-based architecture [1]. For HOMER, a behavior is an independent software module that solves a particular problem, such as navigation or face detection. We refer to behaviors interchangeably as modules. Behaviors exist at 3 different levels, as shown in Figure 2. The lowest level behaviors interface with the robot's sensors and actuators, relaying commands to the motors or retrieving images from the camera. These are described more fully elsewhere [7]. Behaviors at the middle level can be grouped in two broad categories: mobility and human-robot interaction (HRI). Mobility modules perform mapping, localization and navigation [11, 14], as well as resource (battery level) monitoring. HRI modules are for people finding and tracking, speech synthesis and recognition, and facial expression generation [5, 7]. Middle level modules interface with the lowest level through a shared memory mechanism. Each middle level module computes some aspect of the state of the environment. For example, the localization module computes the current location of the robot with respect to the robot's internal maps. These outputs are typically reported to the highest level modules. Each module further offers a set of possible actions the module can effect. Communication among middle and high level behaviors is done using network sockets.

There are four high-level modules: a manager and three planners. The manager maps between the output of the modules and the inputs to the three planning engines. The current outputs of all the modules (and the manager's state) are the current state of the robot, and it is divided up into different sections for each planner. The manager's job is to integrate all the state information from the modules with its own, and present the result to the planning engines. The planning engines together implement the MS-MDP framework to select the actions to perform given the current state. The manager then delegates the actions sent by the planners to the appropriate modules.

4 Management and Planning

The global state of the system is described by a vector $\mathbf{X} = \bigcup_{i=1}^{N} \mathbf{X}_i$, where $\mathbf{X}_i$ is the state vector for each module and N is the number of modules. The manager synthesises this state vector by collecting information from each module, possibly compressing the probabilistic belief states reported by modules by choosing the value with maximum likelihood. In the general case, the planners would take advantage of the information contained in the belief state, by using partially observable MDPs (POMDPs). However, it is P-SPACE hard in the general case to find policies for POMDPs, calling for approximate techniques for robotics applications [10]. Hierarchical methods are another way to combat the complexity of POMDPs [15, 16, 12]. HOMER considers the state vector to be fully observable, and uses MS-MDPs to break the domain into tractable components. After a review of MDPs, we describe our MS-MDP framework and how this is used for the message delivery task.

4.1 Markov Decision Processes

Markov decision processes (MDPs) have become the semantic model of choice for decision theoretic planning (DTP) in the AI community [13]. They are simple for domain experts to specify, or can be learned from data. They are the subject of much current research, and have many well studied properties including approximate solution and learning techniques. An MDP is a tuple $\{\mathcal{S}, \mathcal{A}, \Pr, R\}$, where $\mathcal{S}$ is a finite set of states and $\mathcal{A}$ is a finite set of actions. Actions induce stochastic state transitions, with $\Pr(s, a, t)$ denoting the probability with which state t is reached when action a is executed at state s. $R(s)$ is a real-valued reward function, associating with each state s its immediate utility $R(s)$. Solving an MDP is finding a mapping from states to actions. Solutions are evaluated based on an optimality criterion such as the expected total reward. An optimal solution is one that achieves the maximum over the optimality measure, while an approximate solution comes to within some bound of the maximum.

We use a factored, structured, MDP solver, SPUDD [8], that uses the value iteration algorithm to compute an optimal infinite-horizon policy of action for each state, with *expected total discounted reward* as the optimality criterion. SPUDD uses a representation of MDPs as decision diagrams, and is able to take advantage of structure in the underlying process to make computation more efficient and scalable towards larger environments. The modularity of our system makes representation as a factored MDP simple and typically results in a sparsely connected Markov network. Such sparseness leads to very efficient calculations when using a structured solution approach as in SPUDD. However, if we require simultaneous actions using a single MDP, we need to consider all the possible action combinations, which will imply an additional increase in the size of the state-action space. So we propose a framework for task coordination based on multiple MDPs, that we call *Multiply Sectioned Markov Decision Processes (MS-MDPs)*,

4.2 Multiply Sectioned MDPs

A MS-MDP is a set of N MDPs, all of which share the same goal and state space, but have different action sets. We assume that the actions of each MDP do not *conflict* with the other processes, so that each action set can be executed concurrently with the others. As mentioned in the introduction, we do not find optimal solutions for the global MDP, but simply simultaneously execute the optimal solutions from each sub-MDP. Intuitively, we can think that each MDP is solving one aspect of the global task, coordinated by a common state vector, and in this way accomplish the common goal. Our results for the messenger robot give empirical evidence for this framework. We leave a formal proof of optimality as future work.

Given that we have a factored representation of the state space, each MDP only needs to consider the state variables that directly influence its actions and reward. This implies that each MDP, P_i, will in general have a subset of the state variables spanning its local state space, S_i. Further, we do not consider the effects of combined actions. These two aspects can make a considerable reduction in the action-state of the problem, as we show for the messenger task.

At the design phase, we specify each MDP to solve one aspect of the global task. For HOMER's message delivery task, one MDP can focus on the navigation part, another on the speech dialogue, another on gesture generation. Each MDP can be specified relatively independently of the others, although the goal is the same and the designer should be aware of the other subtasks. Then, each MDP can be solved independently to obtain the optimal policies for each subtask. At the execution stage, all the MDPs are executed concurrently, and the best actions (for each) are selected according to each policy and the current state.

A final consideration is conflicts between the actions of the different MDPs. Conflict in this case is simply a constraint that would preclude two actions from being executed. For example, the robot cannot navigate and recognize people simultaneously because the robotic head must be facing in different directions for each. Currently, we use a simple heuristic to resolve conflicts, but we are working on an extension that includes an *arbiter* to decide the best action based on its value.

4.3 Managing the message delivery task

HOMER's message delivery task consists of accepting messages, finding recipients and delivering messages. In his quiescent state, HOMER explores the environment looking for a message sender. A potential sender can initiate an interaction with HOMER by calling his name, or by presenting herself to the robot. HOMER asks the person for her name (sender), the recipient's name, and the message. During the interactions, HOMER uses speech recognition, speech generation and gesture generation to communicate with people. Once HOMER has a message to deliver, he must find the recipient. This requires some model of the typical behavioral patterns of people within HOMER's workspace. We use a static map of person locations, which is updated when new information is obtained about the presence or absence of persons. This map allows HOMER to assess the most likely location to find a person at any time. Navigation to that location is then attempted. If the location is not reachable, HOMER finds another location and re-plans. If the location is reached, then HOMER attempts to find a potential receiver using face and voice detection. Upon verifying the receivers name, HOMER delivers the message. During the entire process, HOMER will localize in the map if necessary, or it will go *home* to recharge if its battery is low.

The message delivery task can be divided in 3 subtasks, each one controlled by an MDP. The **N**avigator controls the navigation and localization of the robot, the **D**ialogue Manager controls the interaction with people using speech, and the **G**esture Generator controls the interaction with people using gestures performed by an animated face. Each MDP includes the relevant variables as its state space, and controls several behaviors through its actions. The complete state is represented by 13 variables, shown in Table 1.

Variable	Description	MDP
Has message	has a message to deliver	N,D,G
Receiver name	receiver name or none	N,D,G
Sender name	sender name or none	N,D,G
At location	at location of receiver	N,D,G
Has location	has receiver's location	N
Location Unreachable	cannot go to location	N
Receiver Unreachable	cannot find the receiver	N
Battery low	battery is low	N
Uncertain location	uncertain about location	N
Voice heard	detected voice (speech)	D,G
Person close	detected a person	D,G
Called Homer	someone call its name	D,G
Yes/No	yes/no response	D,G

Table 1: Homer's state variables. For each variable, we show the MDPs that include it.

MDP	actions	modules
Navigator	explore	navigation
	navigate	navigation
	localize	localization
	get new goal	location generator
	go home	navigation
	wait	navigation
Dialogue	ask	speech generation
	confirm	speech generation
	give message	speech generation
Gesture	neutral	gesture generation
	happy	gesture generation
	sad	gesture generation
	angry	gesture generation

Table 2: Homer's MDPs and its corresponding actions. For each action, we indicate the modules which effect them.

The actions for each MDP and corresponding behaviors are shown in table 2. Several behaviors do not appear in the table (person detection, voice detection, etc.); these are used to get information by observing the world.

The dialogue manager includes some actions that are *generic*, such as the **ask** action, which can be used to ask for the sender's name, receiver's name and the message. The specific action is determined by the state variables, and it is decided at another level. For the Dialogue MDP, it is like any other action (the transition function considers all the specific actions). We implement these generic actions as finite state machines, although this could be also represented as another MDP in a hierarchical way.

The goal of the message delivery task is encoded in the reward function: a *small* reward for receipt of a message, a *big* reward for message delivery, and a negative reward for a low battery. The Dialogue and Gesture planners only include rewards for message receipt and delivery, while the navigator includes all three.

We solved the 3 MDPs using SPUDD and generated the optimal policies for each one. During concurrent execution of the policies, potential conflicts are avoided by simply giving priority to the Navigator. Thus, if HOMER is navigating to a location, such as *home*, it does not stop for an interaction. Our current work involves using an *arbiter* to resolve these conflicts more generally.

5 Experiments

To validate our approach, we ran several experiments with our robot. Each experiment involves the robot receiving and delivering a message by visiting locations as necessary. Initially, we performed a guided exploration task in order to build all the necessary maps for navigation and localization. We also manually specified a list of possible users and the most likely areas they inhabit. HOMER then ran autonomously for the message delivery task.[1] Figure 3 shows key frames from one message delivery run. Some of the key variable values are shown, as well as the actions taken by each planner (**N**avigator, **D**ialogue, **G**estures). A brief description is also given.

6 Conclusion and Future Work

In this paper, we introduced MS-MDPs as an efficient approach to the design and implementation of socially interactive robots. This technique allows the partition of a robot's task into a number of subtasks, each assigned to an MDP, such that each one can be specified and solved independently. At run-time, all MDPs execute concurrently, coordinated implicitly by common state variables. We validated the MS-MDP framework using it to design our robot HOMER, that approaches the problem of message delivery among people in a workplace.

In the future, we wish to analyze the optimality of MS-MDP policies in order to establish theoretical bounds, and investigate conflict resolutions. Finally, we also plan to extend HOMER's current model such that it can engage in more complex interactions.

References

[1] R. C. Arkin. *Behavior-based Robotics*. MIT Press, 1998.

[1]Videos of three message delivery runs are available at `http://www.ubc.ca/~elinas/homer2.html`.

key frame	variables changed	actions	description
	person close = true voice_heard = true	**N**: - **D**: ask **G**: smile	Person approached. HOMER initiated conversation by asking person's name, smiled encouragement.
	has_message = true has_location = true voice_heard = false sender_name = 4 receiver_name = 1	**N**: navigate **D**: mute **G**: neutral	HOMER received message and recipient, and got location from location planning module. The navigator began moving towards that location. Dialogue manager silent.
	person_close=false uncertain_loc = true	**N**: localize **D**: - **G**: -	During navigation, HOMER's position became uncertain. Navigator MDP localizes. State of other two MDPs not affected by variable *uncertain_loc*.
	at_location=true voice_heard=true yes/no=yes	**N**: wait **D**: deliver message **G**: smile	HOMER reached the target location, detected a person and confirmed the correct recipient (through *yes/no*). Message delivered with smile.
	has_message = false at_location=false battery_low=true	**N**: go home **D**: - **G**: -	HOMER's battery signaled as low, navigator MDP returned HOMER to home.

Figure 3: Example message delivery run, showing key frames, variables that change or are significant, actions taken by each planner (**N**avigator, **D**ialogue, **G**esture), and a description of what happened.

[2] C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, high-level agent programming in the situation calculus. In *Proc. of AAAI '00*, 2000.

[3] W. Burgard, A. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *AAAI '98*, Madison, WI, July 1998.

[4] T. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[5] P. Elinas, J. Hoey, D. Lahey, J. Montgomery, D. Murray, S. Se, and J. J. Little. Waiting with Jose, a vision based mobile robot. In *Proc. of ICRA '02*, Washington, D.C., May 2002.

[6] P. Elinas, J. Hoey, and J. J. Little. Human oriented messenger robot. In *AAAI Spring Symposium on Human Interaction with Autonomous Systems*, Stanford, CA, March 2003.

[7] P. Elinas and J. J. Little. A robot control architecture for guiding a vision-based mobile robot. In *Proc. of AAAI Spring Symposium in Intelligent Distributed and Embedded Systems*, Stanford, CA, March 2002.

[8] J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier. SPUDD: Stochastic planning using decision diagrams. In *Proc. of UAI '99*, Stockholm, 1999.

[9] N. Meuleau, M. Hauskrecht, K.-E. Kim, L. Pashkin, L. P. Kaelbling, T. Dean, and C. Boutilier. Solving very large weakly coupled Markov decision processes. In *Proc. of AAAI '98*, 1998.

[10] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. In *Proc. of AAAI '02*, Edmonton, Canada, 2002.

[11] D. Murray and J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8:161–171, 2000.

[12] J. Pineau, G. Gordon, and S. Thrun. Policy-contingent abstraction for robust robot control. In *Proc. of UAI '03*, pages 477–484, Acapulco, Mexico, August 2003.

[13] M. L. Puterman. *Markov Decision Processes.* Wiley, New York, NY., 1994.

[14] S. Se, D. Lowe, and J. J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant landmarks. *Itnl. J. of Robotics Research*, 21(8):735–758, August 2002.

[15] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of IJCAI '95*, pages 1080–1087, Montreal, Canada, 1995.

[16] G. Theocharous, K. Rohanimanesh, and S. Mahadevan. Learning hierarchical partially observable Markov decision process models for robot navigation. In *Proc. of ICRA '01*, Seoul, Korea, May 2001.

[17] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: A second-generation museum tour-guide robot. In *Proc. of ICRA '99*, Detroit, MI, May 1999.