Hierarchical Factored POMDP for Joint Tasks: Application to Escort Tasks

Fabio-Valerio Ferrari, Abdel-Illah Mouaddib
GREYC - UMR CNRS 6072 - University of Caen Basse-Normandie
Boulevard Marechal Juin, BP 5186, 14032 Caen Cedex, France
fabio-valerio.ferrari@unicaen.fr, abdel-illah.mouaddib@unicaen.fr

Abstract

The number of applications of service robotics in public spaces such as hospitals, museums and malls is a growing trend. Public spaces, however, provide several challenges to the robot, and specifically with its planning capabilities: they need to cope with a dynamic and uncertain environment and are subject to particular human-robot interaction constraints. A major challenge is the Joint Intention problem. When cooperating with humans, a persistent commitment to achieve a shared goal cannot be always assumed, since they have an unpredictable behavior and may be distracted in environments as dynamic and uncertain as public spaces, and even more so if the human agents are customers, visitors or bystanders. In order to address such issues in a decision-making context, we present a framework based on Hierarchical Factored POMDPs. We describe the general method for ensuring the Joint Intention between human and robot , the hierarchical structure and the Value Decomposition method adopted to build it. We also provide an example application scenario: an Escort Task in a shopping mall for guiding a customer towards a desired point of interest.

1 Introduction

The deployment of multi-modal robots in public spaces, such as a mall, for service or human assistance tasks is a growing application field in Robotics. There are already several examples of service robots applied in malls, hospitals, museums and so on, such as in (Kanda et al. 2009). It is in the interest of public and private administrations to make their public spaces easier to use, friendlier to visitors and safer to an increasing elderly population and to citizens with disabilities. However, autonomous robots have to face several issues in order to perform their task in crowded public spaces. First, the environment is dynamic and unpredictable, since it is populated by many individuals, or even a crowd, whether they are workers, customers or bystanders. In addition, robots need to interact and usually cooperate with humans to achieve some kind of task, initiated by the human itself most of the time. This kind of cooperation, however, has a weaker level of commitment with respect to other application fields such as, for example, cooperation with professional workers in an industrial environment.

The European project COACHES (Cooperative Autonomous Robots in Complex and Humans Environments)[1] aims specifically at providing solutions to those issues. Among the challenges faced are knowledge-based representation of the environment, recognition and estimation of human activities and intentions, distributed decision making and planning, and multi-modal human-robot interaction. Specifically, the project's application scenario is the deployment of a team of service robots in a shopping mall which will welcome, assist and guide customers. A mall in the city of Caen, France has been chosen as test-bed.

Our work addresses in particular the decision-making and planning aspects and the theoretic problems associated. We need to take into account the uncertainty of the environment in our model, to perform planning under human interaction constraints, and to adopt a decentralized approach for the cooperation of the whole team of robots. We plan on using Partially Observable Markov Decision Processes (POMDPs), which are models capable of efficiently perform decision-making under uncertainty, with a novel framework for addressing the particular human-robot interaction problems.

The article is structured in the following way:
In Section 2 we address the Joint Intention problem and review Factored and Hierarchical POMDPs.
In Section 3 we describe the proposed hierarchical framework and the associated value decomposition method.
In Section 4 we present an application scenario: an Escort Task of customers in a shopping center.
In Section 5 we describe the algorithms and an early implementation of the model.
In Section 6 we highlight the major points of interest of the proposed model.

2 Background

2.1 The Joint Intention Problem

In cooperative multiagent settings, it is usually assumed that all agents sharing a common task have a persistent commitment towards the achievement of their goal. This assumption, however, cannot always hold, especially within Human

[1]The project is funded by Chist-ERA, 2014 – 2017. http://www.chistera.eu/projects/coaches

Robot Interaction (HRI) scenarios. Humans may have unpredictable behaviors and may drop their commitment to the shared task, because they have changed their mind or compelled by external factors. This is especially true in service robotics applications in public spaces, such as a shopping mall, where human agents are mostly customers passing by and are more prone to change their behavior with respect to a professional worker, and where shops make effort to draw their attention. A behavioral change in the human agent, however, may be caused by different situations. The human may be dropping his commitment to the task, or he may be in need of help, or he may have found an unexpected way to achieve the shared goal. The cooperative agent should be able to tell the difference between these situations, try to understand which is the human's current *mental state* and react accordingly. In other words, the agent should understand *whether* the human is cooperating or not, and *why*.

Joint Intention (JI) Theory (Cohen and Levesque 1991) (Kumar et al. 2002), is a formalism used to define the mental states assumed by agents during teamwork. This formalism has already been used in robotic applications to improve the cooperation level between heterogenous agents, as in (Alami et al. 2006).
JI is defined as a joint commitment to perform a collective action while in a certain shared mental state. In the case of multi-agent systems with a centralized control, JI is trivially always ensured. However, in the case of decentralized, heterogenous systems with partial or no mutual knowledge, as in HRI scenarios, it is important to formalise the cooperation schemes between the agents. JI is what binds the team members together and makes the difference between a functional team and a disfunctional one.

JI Theory is defined formally using a modal language based on first order logic, with the addition of temporal operators and propositional attitudes. We will now review only the basics of JI and the notions most pertinent to our work. For a complete description of all mental state definitions, please see (Kumar et al. 2002).

Let i be an agent in a team A.
Given a goal proposition p and an irrelevance clause q, we can define:
Belief: $B_i(p) =$ agent i believes p holds true.
Desire: $D_i(p) =$ agent i desires p to hold true.
Mutual Belief: $MB_A(p) =$ mutual belief of p by each agent belonging to A.
Mutual Goal: $MG_A(p) = D_i(p) \wedge MB_A(D_j(p)) \quad \forall i, j \in A$.
$\Diamond$ and $\Box$ are temporal logic operators. q is an "escape clause" which allows the agent to relativize its goal and drop the commitment when it comes to believe it to be true. If this clause did not exist, the agent would never drop a commitment freely once adopted. We can also define the Termination Conditions for goal p with respect to q as:
$TC(p, q) = p \bigvee \Box\neg p \bigvee \neg q$
That is, the conditions are true when goal is either achieved (p is true), impossible (p will never be true) or irrelevant (q is false).

We can define the Weak Achievement Goal (WAG) mental state as:

Agent $i \in A$ has a $WAG_{i,A}(p, q)$ if

$$(B_i(\neg p) \wedge D_i(\Diamond p)) \vee (B_i(TC) \wedge D_i(MB_A(TC(p, q))))$$

Which means that agent i has a desire to achieve its goal, and, whenever it believes it to be terminated, it has a desire to ensure that all team-mates share the same belief that the task is terminated.

Two agents i and j in A jointly intend to do an action (or actions) a if they have a joint commitment to doing the action a mutually believing throughout the action execution that they are jointly doing that action as a team (Cohen and Levesque 1991). For a team to have a JI about a, the following conditions must hold:

- $MG_A(\Diamond p)$
- $WAG_{i,A}(p, q) \quad \wedge \quad MB_A(WAG_{i,A}(p, q))$ until $(MB_A(TC(p, q))$
- $(MB_A(DOINGa)$ until a is done.

We use JI Theory as key source of inspiration for ensuring that the human is cooperating with the robot through the whole task execution and mutually believe so.

2.2 Partially Observable Markov Decision Processes

Partially Observable Markov Decision Processes (POMDPs) (Sondik 1971) (Kaelbling, Littman, and Cassandra 1998) are defined as a tuple $< S, A, T, O, R, \omega >$, where:
S is a discrete set of states s.
A is a disrete set of actions a.
$T : S \times A \mapsto \Pi(S)$ is a Transition probability function.
O is a set of observations o.
$R : S \times A \mapsto \Re$ is a Reward function.
$\omega : S \times A \mapsto \Pi(O)$ is an Observation probability function.

The Transition function $T(s, a, s')$ gives the probability to move to state s' when action a is performed in state s. The Reward function $R(s, a)$ gives a real-valued reward (or cost) for performing action a in state s. The Observation function $\omega(s, a, o)$ gives the probability of observing o when action a is performed in state s. Solving a POMDP essentially means finding a policy π, that is, a list of state-action pairs, which maximizes the expected total reward

$$E\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big]$$

where γ is a discount factor and r_t is the reward obtained at time t.

In a Factored POMDP (FPOMDP), the state space is generated by the product of discrete state variables.
Let $X_1, ..., X_n$ be the state variables of the model. Let $Dom(X_i)$ denote the domain of the i-th state variable, and v its values. We denote by x_i the current value of X_i at any given time step, with $x_i \in Dom(X_i)$. The current state of the model can be then represented as a vector of instantiates of all state variables: $\mathbf{x} = (x_1, ..., x_n)$. If we denote by $\mathbf{X}$ the set of all possible values of $\mathbf{x}$, that is, all possible combinations of instantiates of the state variables, then the state-space of the FPOMDP will be $S = Dom(\mathbf{X})$.

The advantage of using FPOMDPs is the possibility to exploit dependencies and independencies of state variables in the transition probabilities. For example, a state variable X_i may only depend on its previous value, regardless of the value of other variables: $Pr(x'_i|x_1, ..., x_n) = Pr(x'_i|x_i)$.

2.3 Hierarchical POMDPs

Despite being a powerful tool for decision-making and planning, using POMDPs in real-world applications has a notable drawback: the problem quickly becomes intractable as the size of the state-space grows exponentially. Additional frameworks and methods must be adopted in order to reduce the state-space during planning. One efficient solution is to use a Hierarchical structure for POMDPs.

Among the possible approaches, one method is to exploit the structure of the domain with a decomposition based on actions (Pineau, Roy, and Thrun 2001). For instance, the overall task can be divided into several sub-tasks. Lower levels of the hierarchical structure implement and solve each sub-task, while upper levels plan to choose which sub-task to perform. It is based on the assumption that not all state information is relevant for the completion of the current sub-task. Using only the state variables necessary for the current sub-task results in a reduction of the state-space during planning. In our work we adopt a sub-task decomposition to better ensure the JI between the human and robot agents. We make a distinction between sub-tasks which proceed into achieve the objective of the overall task and those with the aim of "repairing" the JI whenever the cooperation is missing or weak.

3 Proposed Framework

We will now describe our proposed framework: a hierarchical factored POMDP for decision-making in cooperative HRI domains. The hierarchical structure is constructed using both a sub-tasks decomposition and a Value decomposition, which is a novel method of abstraction developed specifically for addressing the JI problem of joint tasks in HRI domains. Using such method, the state abstraction from lower levels to the uppermost highlights the essential components of the JI.

3.1 The Joint Intention set

The main idea for dealing with the JI problem is the following.

Given the joint state-space S, we define a set of states $JI \subset S$ where the JI is preserved, that is, where all agents have a commitment to the common task, share a Weak Achievement Goal mental state, and are actively performing actions to achieve the task while mutually believing so. As long as the current state belongs to JI, then all agents are trying to achieve the common goal: everything is going well and there is no need to react. Otherwise, the agent needs to try to bring the system back to a JI state.

We can consider the joint state-space S as the combination of each agent state-space, namely the human one S_H and the robot one S_R. Therefore, when trying to return to a JI state, we can either attempt to change the robot's state or the human's, or both. Hence, the high-level algorithm of our model consists in choosing, given the current state, the general *course of action* to adopt: either keep staying inside the JI, change the robot's state, or change the human' state. In addition, another course of action is considered: the task termination. As explained in Section 2.1, if an agent believes that the task is terminated, if it has a WAG mental state, then it will try to ensure that the other agents in the team share the same belief. Therefore, the agent cannot end the task abruptly, but it will perform a last sequence of actions to reach an exit state. How the choice of such action course is performed will be described in Section 3.3.

3.2 Hierarchical structure

The proposed model has a three-level hierarchical structure, with Bottom, Mid and Top layers. When creating the model, the Mid level represents the core. We first define the state variables $X_1, ..., X_n$ and the corresponding state-space $S = Dom(\mathbf{X})$, then the other two layers are generated from it through decomposition methods.

Bottom layer The Bottom level takes information from sensors and executes low-level planning, such as robot navigation. Actions taken at this level are *primitive actions*. The Bottom Layer is modeled as a POMDP $< S^{bot}, A^{bot}, T^{bot}, O^{bot}, R^{bot}, \omega^{bot} >$, with S^{bot} being the state-space generated through sub-task decomposition. Given the current sub-task (or macro-action) τ, provided by upper levels, we only use a subset of state variables relevant for that sub-task during planning phase, denoted as $X_\tau \subseteq X$. Therefore, $S^{bot} = Dom(\boldsymbol{X_\tau})$.
O^{bot} is a set of primitive observations coming from the robot sensors and ω^{bot} the consequent observation probability function. A^{bot} contains primitive actions, and T^{bot} and R^{bot} are the associated transition probability function and reward function over S^{bot}.

Mid layer The Mid level, defined as a POMDP $< S, A^{mid}, T^{mid}, O^{mid}, R^{mid}, \omega^{mid} >$ is an intermediate layer. It constitutes the core of the structure, as it describes the main state variables and passes information to both upper and lower levels. Actions taken at this level will be called *macro-actions*.
The Mid level is an abstract layer. It abstracts the state space of the Bottom layer in order to focus on the variables which contribute to define the current mental state of human and robot. It generates plans of macro-actions for both achieving the task goal and re-establishing cooperation when missing. In addition, the Mid level can be considered akin to an observer: it acts as an interface between the layers and provides them the information necessary to generate their plans.
Here, observations in O^{mid} may be either directly primitive observations, or information on the current state coming from the Bottom layer. Similarly, A^{mid} may consist of either macro-actions or primitive actions with a direct effect on the Bottom state. Transition, Reward and Observation functions $T^{mid}, R^{mid}, \omega^{mid}$ are computed accordingly.

Top layer The Top level implements the general algorithm for solving the JI problem presented in Section 2.1. The model consists of a factored POMDP $< S^{top}, A^{top}, T^{top}, O^{top}, R^{top}, \omega^{top} >$, with binary variables $Z_1, ..., Z_n$ such that $S^{top} = Dom(\boldsymbol{Z})$.
The purpose of the Top layer is two-fold. First, the output actions taken at this level are the *sub-tasks* previously mentioned, which define the general *action courses* to be taken and reduce the planning load of lower levels (Figure 1). As the mental state of agents may be caused by a variety of factors, the Mid layer may present several state variables and consequently an obstacle for the analysis of the current level of cooperation. Therefore the Top level provides a great help for the *diagnostic* of the current state: the binary variables represent in a very concise way the factors which define the JI and human-robot cooperation. It is possible to assess immediately whether the agents are in a JI state and, if not, know the reason why.
The state-space of the Top layer is generated using the Value decomposition method, which is described in the next Section.
At Top level, observations in O^{top} come from the Mid level. In such way, uncertainty and state belief are propagated from the lower layers to the uppermost. A^{top} consists of a set of sub-tasks. Given the high degree of abstraction, functions T^{top}, R^{top} and ω^{top} should be straightforward.

3.3 Value decomposition

In addition to sub-task decomposition, another method is used to generate the state-space of the Top layer. We will describe the method for an MDP, describing the effects on states, but we believe that the same reasoning may be applied on observations in the case of a POMDP.
We perform a value decomposition of variables at Mid level: for each variable, we divide its domain in two sets, those values which may admit a JI state and those who don't. Remark that as the definition of which states belong to the JI set is application-dependent, so is the possibility to make such kind of decomposition, but we expect that its existence can be assumed in most HRI applications.
Specifically, we assume that, for at least one state variable X_i, $\exists$ a value $v \in Dom(X_i)$ s.t. $\{\mathbf{x}|\mathbf{x}_i = v, \mathbf{x} \in JI\} = \emptyset$. Then, for such variables we partition its domain $Dom(X_i) = (D_i^T \cup D_i^F)$, with D_i^F being the set of values that never appear in a JI state, and D_i^T the set of values that appear in at least one JI state. Remark that values in D_i^T *may* admit JI, but do not necessarily imply it.

$$D_i^F = \{v|\{\mathbf{x}|\mathbf{x}_i = v, \mathbf{x} \in JI\} = \emptyset\}$$
$$D_i^T = \{v|\{\mathbf{x}|\mathbf{x}_i = v, \mathbf{x} \in JI\} \neq \emptyset\}$$

Then, we can define boolean variables which simply describe in which domain set the current variable belongs to.
Let $Z_1, ..., Z_n$ be the state variables of Top level, and $\mathbf{z} = (z_1, ..., z_n)$ the vector of their current values, with $z_i \in Dom(Z_i) = \{\top, \bot\} \quad \forall i$.
Each variable is defined as:

$$z_i = \begin{cases} \top, & \text{if } x_i \in D_i^T \\ \bot, & \text{otherwise} \end{cases}$$

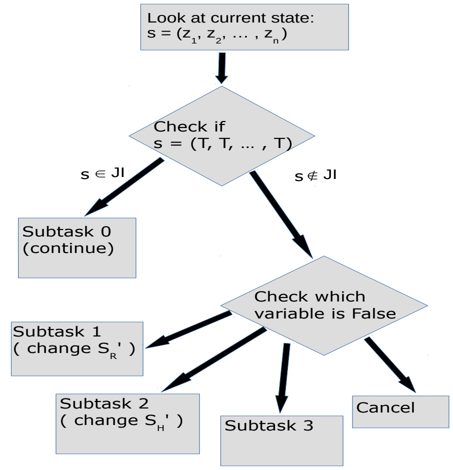


Figure 1: General Top layer algorithm

Therefore, if $\mathbf{z} = \{\top, \top, ..., \top\}$, then the state $\mathbf{z} \in JI$. Otherwise, the JI is not ensured and the agent needs to choose an appropriate sub-task to change its state.

This approach allows us to implement the general algorithm described in Section 3.1. The definition of binary variables resulting from the value decomposition enables an efficient and compact planning on a Factored POMDP. In addition, this decomposition has the additional benefit of giving more information on which state variable is preventing the JI. By looking at the current state at Top level, if variable z_i is false, we know *why* the state is not in the JI, and *where* we should act: on variable X_i. Such reasoning is adopted for determining the next sub-task (See Figure 1).

Remark, however, that information of current state is incomplete at Top level. For instance, at Top level, state variable Z_i only tells if x_i belongs to D_i^T or not, but doesn't know the exact value of x_i. We need to look into the Mid level to have a more accurate representation of the state. On the other hand, the state-space of Bottom level is only a subset of S in Mid level. The state variables used depend on the current sub-task or macro-action. Also, the goal state and transition probabilities depend on the current state at Mid level, and transition probabilities of Top layer depend on the current state at lower levels. For all these reasons the Mid layer has an important role in providing the necessary informations to the other levels.

Depending on the application domain, a single decomposition is not sufficient to check the presence of JI. There may be some states which should not belong to the JI but are seen to do so anyway at the Top level. This is due to the fact that values in D_i^T do not necessarily imply the JI: some values may negate the JI only when coupled with particular values of other variables. In such case, a second step is required to generate other binary variables. To do so, we only consider values in D_i^T and check if they may be "dangerous".
A value $v \in Dom(X_i)$ is critical iff $\exists v' \in D_i^T$ s.t. $\{\mathbf{x}|\mathbf{x}_i = v, \mathbf{x}_j = v', \mathbf{x} \in JI\} = \emptyset$, with $i \neq j, v \neq v'$.
We can then define other variables at top level Z_i', which

hold *true* iff x_i is *not* critical and false otherwise.

Another interesting knowledge provided by Value decomposition is Variable Relevance. As stated in Section 3.2, only a subset of the state variables are relevant during the planning phase, which allows for a reduction of the state-space and thus more efficient planning. While designing the model, we check the values which allow the JI and those who don't. If a variable X_i always allows JI, that is, if $D_i^F = \emptyset$, then that variable is irrelevant for the planing phase: regardless of its value, z_i will always be True, so it won't provide any meaningful information for choosing the appropriate sub-task to execute. On the opposite, if $D_i^T = \emptyset$, the variable will never allow JI, meaning that the problem, as currently modeled, does not admit a solution.

4 Escort Task

We will now describe an application scenario of the proposed framework. This application is part of the COACHES project. The project aims at deploying a team of mobile assistance robots in a shopping mall, for helping customers find a navigate to points of interests.

The robots will be built on a Segway mobile base and equipped with several sensors, including laser range finders, RGBD cameras and microphone as well as a touchscreen interface, audio speakers and speech recognition and synthesis for multi-modal interaction with the human. In the shopping mall, the robots will have several tasks to perform:

- **Assistance**: provide information and instructions to visitors, display the path towards a chosen point of interest.
- **Escort**: physically lead the visitor to a desired location. The visitor and the robot must perform a joint task.
- **Security**: detect landmarks (eg. wet floor signs) and secure the area.
- **Advertisement**: "smart"and context-based, depending to the nearest shop and the customer it is interacting to.

Among these, however, we will focus on the Escort Task. When a customer asks assistance for reaching a point of interest, the robot offers to physically guide and escort him along the way. In such task, the robot does not only have to lead the human towards a goal position, but also adapt to its behavior and ensure that the commitment to the joint task (that is, reaching the location) is preserved, or at least understand when it is dropped. For example, while moving, the human may take a different path than the one chosen by the robot. The human may also temporarily stop, or change direction, or turn around etc. It may be because he has changed his mind and does not want to reach the goal anymore, or because he is lost, or just slightly distracted, or because he knows an alternative way to reach the location. A strict leader-follower paradigm is not sufficient, since we want the robot to adapt to the human behavior and decide whether "meet him halfway" or not.

4.1 State variables

In order to estimate the current behavior of the human and try to understand its intentions, we use the robot's sensors (mainly a rear camera) to observe two features: the *interaction distance* and the *attention level* of the human.
Interaction Distance is the relative distance of the human with respect to the robot. We use the studies on Proxemics to define a set of interaction distances. Proxemics (Hall 1966) have already been applied in robotics to provide more sociable and human-friendly navigations and interactions in public spaces (Pandey and Alami 2009). In our work, their main use is to help in better understanding the human's behavior: if the human stays too far from the robot, it may mean that he needs help or that he is going away, while if he wanted to follow the robot he would probably stay within the *Personal space* distance. The four distances defined by Proxemics are the following:

- Intimate I : between 0 and 46 cm.
- Personal P : between 46 and 122 cm.
- Social S : between 1.2 and 3.7 m.
- Public U : between 3.7 and 7.6 m. (and beyond)

In addition to the distance, also the Relative Position of the human with respect to the robot is taken into account: the human may be situated on the Front, Rear, Left or Right side of the robot.

The Attention Level, instead, can be estimated by performing some gaze detection and activity recognition processes on the rear camera data and trying to determine how much the human is concentrated on following the robot.
The Attention level consists of the following values:

- F Focused: the human's attention is focused on the robot.
- D Distracted: the human is slightly distracted. He may be looking at the shops nearby, or at his phone, or similar cases.
- L Lost: the human is completely neglecting the robot. He may be turning back, or concentrated on some activity other than the joint task.

Following the concepts described in Section 2.1, the *Task status* captures the knowledge of the agent about whether the Termination Conditions hold true or not, and more in general about the status of the task. The task may be:

- On Ongoing: the goal location has not been reached yet, but still can be.
- Arr Arrived: the goal location has been reached.
- Imp Impossible: the goal can never be achieved. The notion of irrelevance clause is also included in this state.

Both human and robot have their own Task Status variable. The status as seen by the robot is supposed to correspond to the actual state of the task. The necessity of ensuring a mutual belief of the task status, however, requires a separate variable for the human.

The overall Escort Task is to be considered successfully achieved when both human and robot have $TaskStatus = Arr$, and, of course, they are actually both at the desired location.

Therefore, the main state variables in our model are the following:

- Interaction distance $Dist$
- Attention level Att
- Relative position $Side$
- Robot task status R_{task}
- Human task status H_{task}

Now that the state variables are defined and able to generate the whole state-space, we can define the JI set. Obviously, a $Lost$ Attention level means that the human is not cooperating with the robot, and thus negates the JI. Values $Intimate$ and $Public$ of the Interaction Distance indicate a distance which is respectively too near and too far for a good cooperation. The $Distract$ and $Social$ values are considered "critical". If the human is slightly distracted while following the robot, the state is considered acceptable and belongs to the JI set, since we cannot pretend to have constant attention, but only as long as he is not too far from the robot (that is, at $Personal$ distance). If the human is both $Distract$ and at $Social$ distance, then cooperation is not ensured.

Some examples of situations belonging or not to the JI are shown in Figure 2. The examples shown correspond to the following states:
a) $x = (P, F, Rear, On, On) \in JI$
b) $x = (P, D, Rear, On, On) \in JI$
c) $x = (S, L, Rear, On, On) \notin JI$
c) $x = (U, F, Rear, On, On) \notin JI$

Then, for each variable we define which values may or not admit the JI. Specifically:
$D^F_{Dist} = \{I, U\}$; $D^T_{Dist} = \{P, S\}$
$D^F_{Att} = \{L\}$; $D^T_{Att} = \{F, D, A\}$
$D^F_{Side} = \{Front\}$; $D^T_{Side} = \{Rear, Left, Right\}$
$D^F_{R_{task}} = \{Arr, Imp\}$; $D^T_{R_{task}} = \{On\}$
$D^F_{H_{task}} = \{Arr, Imp\}$; $D^T_{H_{task}} = \{On\}$

Hence, we can build the binary variables of Top layer: $Z = (Z_{Dist}, Z_{Att}, Z_{Side}, Z_{R_{task}}, Z_{H_{task}})$

4.2 Actions, Observations and Rewards

Primitive actions are divided in two main groups: navigation and dialogue actions. Navigation include *Move* actions ($North$, $East$, $South$, $West$), *Turn* actions ($TurnLeft$ and $TurnRight$) and the $Pause$ action, which does nothing. Dialogue actions include $DrawAttention$, where the robot tries to revert the human to a $Focus$ state, and $CheckStatus$, for obtaining a more confident observation about the $HumanTaskStatus$. When in doubt, the robot can ask the human if he still desires going to the specified location. The answer will provide a great level of certitude about the joint commitment.

Observations are drawn from the sensors of the robot, usually after an elaboration process to semantically interpret the data. For instance, scene recognition, gaze detection and human behavior estimation techniques may be applied on the cameras to try to classify the attention level of the human. Observations may also come from the multi-modal interaction capabilities. This is the case of the observation for the *Human Task status* resulting from the $CheckStatus$ action. In our model, we consider all variables to be observable.

Rewards are of three types. A standard unitary cost of -1 is given for most actions, except for $Pause$. A huge cost (-50) is given to states with an $Intimate$ distance to prevent collision with the human. Lastly, a reward (+100) is given to the goal state, depending on the current sub-task.

The sub-tasks themselves are mainly four:

- Continue: the robot focuses on moving towards the desired point of interest.
- Navigate: the robot moves in order to re-establish the JI whenever the human is too far or too close.
- Dialogue: the robot interacts with the human to change its level of attention or the mutual knowledge of the task status.
- Cancel: before aborting the task, the robot must attempt to ensure the mutual knowledge that the task will be terminated.

5 Algorithms

The implementation of the proposed model is still a work in progress. Here we describe a preliminary version, in which the state of the system is assumed to be fully observable. We will also focus on the Navigation subtask for keeping the Distance and Position of the human within the JI.

5.1 Bottom layer

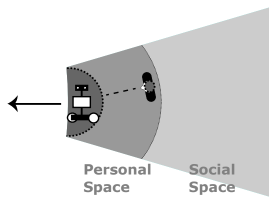
The bottom layer is implemented as a simple grid world domain with discrete coordinates. Coordinates are centered on the robot, so that only the position of the human and the goal location are variable. The main purpose of the bottom layer is to provide observations and contribute in generating the transition probabilities of the mid level.

The domain of the bottom layer is divided in zones Z, which correspond to the proxemic distances (Intimate, Personal, Social and Public). By checking the distance between the human's and robot's position in each state of the bottom level, we define N_Z as the number of states belonging to each Zone. In addition, each zone is divided in three regions: a border region $F_+(Z)$ with the successive zone, one $F_-(Z)$ with the previous zone, and a middle region $F_0(Z)$. Each border is defined as the region of a zone where a single discrete step is sufficient to transitate into another adjacent zone. Trivially, the Intimate zone does not have a border with a successive zone, while the Public one does not have a border with a previous zone. By counting the number of states $N_{F(Z)}$ belonging to a particular zone region, we can compute the conditioned probability of the human position x to be in said region as

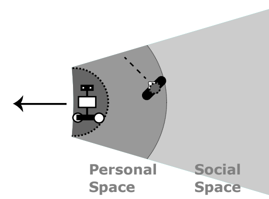
$$P(x \in F(Z) | x \in Z) = \frac{N_{F(Z)}}{N_Z}$$

with $F(Z) \in \{F_+(Z), F_0(Z), F_-(Z)\}$.

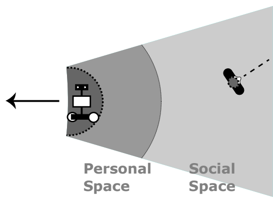
These probabilities will be used to compute the probability of transition from one zone to the other at the intermediary level. In such way, the exact position of the human is not known, and the transition probabilities will be less precise,



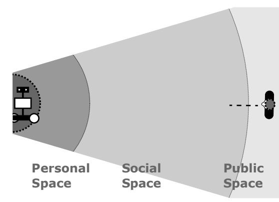
(a) Focus, Personal



(b) Distract, Personal



(c) Lost, Social



(d) Focus, Public

Figure 2: Examples of Interaction distance and attention level. a) This case represents the ideal situation for cooperation in the Escort task. b) As long as the distance is not too far, a slightly distracted behavior is acceptable. c) The human probably is either dropping the commitment or requires assistance. d) The human is very far, but focused on the robot, meaning probably that he is still cooperating, but needs to catch up with the robot.

but the abstract mid layer will be completely independent from the bottom layer during the planning phase. Once the probabilities are computed, no other information from the bottom layer is required to generate the policy.

5.2 Mid layer

The intermediate level is divided into several sub-tasks, each implemented as an independent MDP, capable of generating a policy within a reduced set of state variables, actions and with rewards specific for the sub-task.

The transition probabilities for the Interaction Distance between human and robot are computed by taking into account both the human and robot movement. The human movement is given by three possible behaviors: $MoveToRobot$, where the human moves one step in the direction of the robot, $Deviate$, where the human moves one step in a random direction, and $Stay$, which is no motion at all. The probability of each behavior depends on both the current Attention Level and Interaction Distance, as well as the robot's action. These three behaviors are exclusive. The probability of transition from one zone to the successive, considering the contribute of the human movement only, is

$$P(H_{succ}|Z) =$$
$$P(F_+|Z)(P(MoveToRobot|Z) + \varphi P(Deviate|Z))$$

With the notation simplification
$P(F_+|Z) = P(x \in F_+(Z)|x \in Z)$ etc.
$\varphi \in [0, 1]$ is a constant factor indicating the probability, among the possible casual directions of $Deviate$, that the chosen direction is heading towards the robot. It is also used for opposite directions for specularity. In a grid world domain, there are 8 possible directions, and φ is set to $\frac{3}{8}$. Similarly, the probability of transition to the previous by the human is

$$P(H_{prev}|Z) = P(F_-|Z)\varphi P(Deviate|Z))$$

and the probability to stay in the same zone is simply the remaining probability since the three possible motions are exclusive.

$$P(H_{same}|Z) = 1 - P(H_{succ}|Z) - P(H_{prev}|Z)$$

Regarding the contribute of the robot's movement, the probabilities are computed simply by checking if the current action of the robot is heading towards or away from the current position of the human, which is independent from the current zone, and conditioned on the probability of being in border regions:

$$P(R_{away}|Z) = P(Away)P(F_-)$$
$$P(R_{towards}|Z) = P(Towards)P(F_+)$$
$$P(R_{stay}|Z) = 1 - P(R_{towards}|Z) - P(R_{away}|Z)$$

The complete probabilities are then computed:

$$P(x' \in Z + 1|x \in Z) =$$
$$P(H_{succ}R_{towards} \cup H_{succ}R_{stay} \cup H_{same}R_{towards})$$

$$P(x' \in Z - 1|x \in Z) =$$
$$P(H_{prev}R_{away} \cup H_{prev}R_{stay} \cup H_{same}R_{away})$$

$$P(x' \in Z - |x \in Z) =$$
$$P(H_{same}R_{stay} \cup H_{prev}R_{towards} \cup H_{succ}R_{away})$$

Other transition probabilities are set with much less amount of computations.

As already mentioned, each sub-task is implemented as an independent MDP. For instance, the Navigation sub-task is described in Algorithm 1. $computeT(S^{bot}, s, a)$ is the procedure computing the transition probabilities as explained in this section. In the current implementation, actions at the Mid layer have a direct correspondence with primitive actions, so that the resulting policy π directly translates to a policy of primitive actions.

Data: Bottom statespace S^{bot}
Result: policy π
$S \leftarrow BuildStates(X_{Dist}, X_{Att}, X_{Side})$;
forall the *action* $a \in A^{mid}$ **do**
  forall the *input state* $s \in S$ **do**
    forall the *output state* $s' \in S$ **do**
      $T^{mid}(s, a, s') \leftarrow computeT(S^{bot}, s, a)$;
    end
  end
end
$goalState \leftarrow (P, F, Rear)$;
$\pi \leftarrow ValueIteration(S, A^{mid}, T^{mid}, R^{mid})$;
return π

Algorithm 1: Navigation sub-task algorithm

5.3 Top Layer

The top layer implements the value decomposition method described in Section 3.3. First, it takes the current state at Mid level and checks for the values which negate the JI and also those labeled as critical (defined in Section 4.1). It then chooses whether to start a Navigate or Dialogue sub-task to re-establish the JI, based on which variable negates it: in case of the $Dist$ variable, the Navigate sub-task will try to move towards or away from the human, while if the human's Attention level is *Lost*, the Dialogue sub-task will try to draw his attention and change it to *Focus*.

Data: Mid state-space **X**
Result: policy π
Z $\leftarrow ValueDecomposition($**X**$)$;
forall the $z \in$ **Z** **do**
  if $z = \{\top, \top, ..., \top\}$ **then**
    $\pi \leftarrow$ Continue subtask;
  else if $z_{Dist} = \perp$ **then**
    $\pi \leftarrow$ Navigation subtask;
  else if $z_{Att} = \perp$ **then**
    $\pi \leftarrow$ Dialogue subtask;
  else
    Abort task;
  end
end
return π

Algorithm 2: Top level algorithm

6 Conclusion and Future Work

We have presented a novel approach to address the problem of ensuring cooperation between an human and a robot agent within a collaborative task. We have described the hierarchical structure, based on Factored Partially Observable Markov Decision Processes, focusing on the Value Decomposition method developed for building the variables of the Top layer. We have also described an example application of human-robot cooperation, an Escort Task for a service robot in a shopping mall. This example shows how the JI can be seen as result of several state variables and how acting on the corresponding values contributes to keeping or re-establishing the cooperation between the agents.

While both JI Theory and POMDPs have been already applied in Literature for human-robot cooperation scenarios, we believe that few work has been performed in trying to integrate the two. The presented framework would therefore provide a contribute in that sense.

In addition, we believe that an interesting point of our framework is the diagnostic potential of Value Decomposition: the method presented allows to extract and express in a concise way all the relevant information about the JI and the cooperation level of the agents, allowing not only to establish whether the JI is preserved, but also understand the reason why it is not.

We plan on developing the implementation of the model, which is currently at a preliminary stage. We then plan to test the model at the test-bed shopping center, and to extend the framework to a multi-agent setting with a decentralized approach.

References

Alami, R.; Chatila, R.; Clodic, A.; Fleury, S.; and Montreuil, M. H. V. 2006. Towards human-aware cognitive robotics. In *In Cogrob 2006, The 5th International Cognitive Robotics Workshop (The AAAI-06 Workshop on Cognitive Robotics*.

Cohen, and Levesque. 1991. Teamwork. *Nous, Special Issue on Cognitive Science and AI.*

Hall, E. T. 1966. The hidden dimension.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *ARTIFICIAL INTELLIGENCE* 101:99–134.

Kanda, T.; Shiomi, M.; Miyashita, Z.; Ishiguro, H.; and Hagita, N. 2009. An affective guide robot in a shopping mall. In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, HRI '09, 173–180. New York, NY, USA: ACM.

Kumar, S.; Huber, M. J.; Cohen, P. R.; David; and Mcgee, R. 2002. Toward a formalism for conversation protocols using joint intention theory. *Computational Intelligence* 18:2002.

Pandey, A. K., and Alami, R. 2009. Towards a sociable robot guide which respects and supports the human activity. In *CASE*, 262–267. IEEE.

Pineau, J.; Roy, N.; and Thrun, S. 2001. A hierarchical approach to pomdp planning and execution. In *Workshop on Hierarchy and Memory in Reinforcement Learning (ICML*.

Sondik, E. J. 1971. The optimal control of partially observable markov processes. Technical report, DTIC Document.