

Transparent Role Assignment and Task Allocation in Human Robot Collaboration

Alessandro Roncone^{*1}, Olivier Mangin^{*1} and Brian Scassellati¹

Abstract— Collaborative robots represent a clear added value to manufacturing, as they promise to increase productivity and improve working conditions of such environments. Although modern robotic systems have become safe and reliable enough to operate close to human workers on a day-to-day basis, the workload is still skewed in favor of a limited contribution from the robot’s side, and a significant cognitive load is allotted to the human. We believe the transition from robots as recipients of human instruction to robots as capable collaborators hinges around the implementation of transparent systems, where mental models about the task are shared between peers, and the human partner is freed from the responsibility of taking care of both actors. In this work, we implement a transparent task planner able to be deployed in realistic, near-future applications. The proposed framework is capable of basic reasoning capabilities for what concerns role assignment and task allocation, and it interfaces with the human partner at the level of abstraction he is most comfortable with. The system is readily available to non-expert users, and programmable with high-level commands in an intuitive interface. Our results demonstrate an overall improvement in terms of completion time, as well as a reduced cognitive load for the human partner.

I. INTRODUCTION

In recent years, research on constructing fully autonomous and self-capable robotic systems has shifted toward partially-capable machines that collaborate with human partners, allowing the robot to do what robots do best and the human to do what humans do best. Among the vast range of strategies employed in human-robot collaborative activities, common approaches are: i) dividing the work into independent sub-tasks to be performed in parallel with little to no overlap—such as in factory lines [1]; ii) exploiting robot-specific benefits—e.g. the absence of fatigue and need to rest—to provide support for the more dexterous and capable human partner [2]. In both cases, a certain amount of autonomy from the robot’s side is extremely beneficial in terms of efficiency and efficacy of the collaboration. Moreover, the degree of transparency about task execution and role assignment that the partners exhibit is crucial: both peers need to reason and to communicate about the role of each partner, so that the other agent can act accordingly.

In the last decade we have witnessed a partial shift in research from observation-based mechanisms for skill

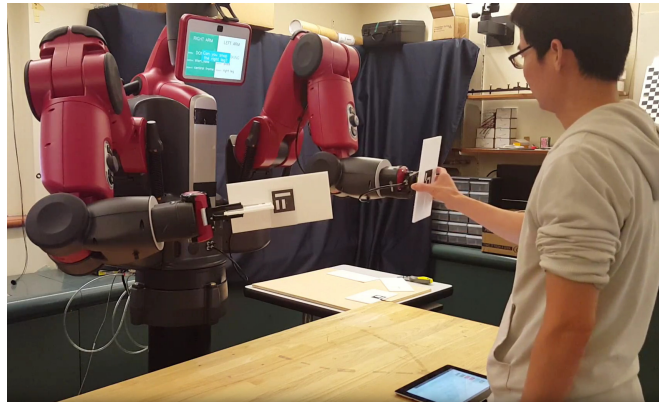


Fig. 1. The experimental setup is composed of the Baxter robot engaging in the joint construction of flat-pack furniture with an human partner (see Section IV-A for details).

acquisition (e.g. learning from demonstration or reinforcement learning) toward collaboration-based mechanisms for task learning [3], [4] and joint operations within a shared workspace [5]. Unfortunately, collaborative robots still lack efficient ways of coordinating their actions with a human peer, as for example another human would do in similar settings [6]. Such capabilities involve on-the-fly work distribution, bi-directional communication about task progress, and adaptive (re-)planning in case of errors or unpredicted human decisions. Additionally, recent work in human-robot teaming suggests how the collaboration benefits from the human partner being in the position of leveraging his superior dexterity or decision making capabilities [7]. Typically, these are domains in which it is hard for the robot to fully understand the state of the world, especially when the human reacts to unforeseen situations for which the robot has not been trained or programmed. In such contexts, the state of the system (composed of the environment as well as the human partner with his intentions and beliefs) is only partially observable and understandable by the robot. Whilst communication may curtail the uncertainty the robot has about the state of the world, there are some fundamental limitations that a constant communication channel between the two peers would entail. First, communication is costly, because it requires both partners to allot time for it, and increases the cognitive load on the human. Further, vocal communication with synthetic systems may be faulty, particularly in noisy environments such as factories or workshops. Finally, communicating about every aspect of a task is often inefficient compared to directly acting, which represents an

This work has been supported by a subcontract with Rensselaer Polytechnic Institute (RPI) by the Office of Naval Research, under Science and Technology: Apprentice Agents.

^{*} A. Roncone and O. Mangin contributed equally to this work.

¹ The authors are with the Social Robotics Lab, Computer Science Department, Yale University, New Haven, CT 06511, USA firstname.surname@yale.edu.

implicit way of communicating about the task itself.

In this work, we set forth to address the following problems: i) the ability of collaborative robots to plan under uncertainty, and specifically in situations where the state of the task as well as the intentions of the human partner are only partially observable, ii) the capability of deciding when to communicate and what to communicate about. We target collaborative robots that are capable of choosing the best execution plan according to the information they are provided with (i.e. a suitable representation of the task), and are able to explicitly optimize some task-relevant parameters (such as completion time) while at the same time reducing the cognitive load on the human partner. Importantly, we focus on implementing a system that is transparent by design, in order for the human to be able to choose the appropriate next action or to timely intervene if the system’s decisions are wrong. Thus, we will treat communication not only as a way for the robot to gather information about the environment, but also as a channel where the robot itself can transfer information to the human partner about its own internal state and intents. The contributions of this work are: i) a fully automated technique able to convert human-readable task models into robot-executable plans; ii) the ability for the system to reason about the task being performed and provide a transparency layer to the user, on top of executing the task; iii) the demonstration of this technology in a real collaborative scenario, where it optimizes the overall task completion time by actively making decisions on role assignment and task allocation.

In the following, we introduce the reader to hierarchical task models and partially observable Markov decision processes (POMDPs, Section II). Then, we describe the proposed approach (Section III), and the experimental scenario used to evaluate the system (Section IV). Finally, results collected from experiments are described in Section V, followed by the discussion and conclusions (Section VI).

II. BACKGROUND AND RELATED WORK

This work rests on the idea that shared knowledge and transparent commitment to the task are central for collaborative behaviors. We base our approach on an original combination of two elements. On one hand, we exploit high level task models such as hierarchical task models, that are understandable by both the human and the robot. On the other hand, we capitalize on adaptive planning from POMDPs, that offer the ability to reason under uncertainty and plan at the high level of representation, a level that is most suitable for communication with the human partner.

A. Hierarchical Task Models

Hierarchical structures form an appealing framework for high-level task representations. In particular, hierarchical task networks (HTNs) have been widely used to describe task representations and derive robotic planners [8]. Another example are “and/or” graphs that represent tasks as hierarchies of conjunctions and alternatives of subtasks; they also provide efficient planning for single and multiple agents [9].

In collaborative setups, hierarchical task models can help predicting the actions of a human peer, hence improving a robot’s ability to provide support in the task [10]. Additionally, their level of abstraction often makes them close to human intuitions about the proceeding and constraints of the tasks, which facilitates human-robot communication about the task. The level of abstraction also enables human operators with moderate to no training to manually provide such task descriptions. The hierarchical nature of these models makes it easy to reuse parts from one task to a similar one. On the other hand, because manual task specification is not always feasible or efficient, active research explores techniques to learn hierarchical task models automatically from demonstration [4] or self-exploration [11].

B. Partially Observable Markov Decision Processes

We use the Markov decision process (MDP) framework to model the optimization of the robot’s actions, an approach that has been shown to provide efficient planning for human-robot interaction (e.g. [2]). In this paper, we additionally optimize the actions related to the task assignment dynamics between the human and the robot. Our design choice of extending the MDP approach to include partial observability with POMDPs has two main advantages. Firstly, it handles the uncertainty in the interaction as non-observability. This includes both misalignment while the human progresses in the task and changes in the environment that are not observable by the robot, e.g. when the human performs a complex manipulation whose progress may be difficult to observe. Indeed, POMDPs and similar models (e.g. MOMDPs) have been shown to improve robot assistance [12] and team efficiency [13] in related works. Secondly, partial observability enables to model the robot’s decisions about communicative actions. These do not directly modify the state of the world but instead gather knowledge. In the following example, the robot decides whether or not to ask the human if his action finished. Similar models have been used for polling the peer’s internal state in a driving setup [14] and communicating in a collaborative tasks [15], [16]. Similarly to [15], in this work we explicitly leverage the partial observability principle to model the internal (not observable) state of the human partner, such as the action he is currently involved with or his beliefs and intents. In doing so, we address both task allocation and information gathering and demonstrate the efficiency of the resulting system in a real robotic experiment.

III. METHOD

Our work leverages recent advancements in the field of high-level task reasoning and representation. Importantly, we focus on improving the efficacy of the collaboration in the context of a task model that is already learned or directly provided by the user. As a consequence of this, we do not direct our efforts on learning task models from demonstrations [3], [17], although we exploit similar representations, as detailed in Section II-A. Our work addresses questions similar to [18], [7], but differs by providing means

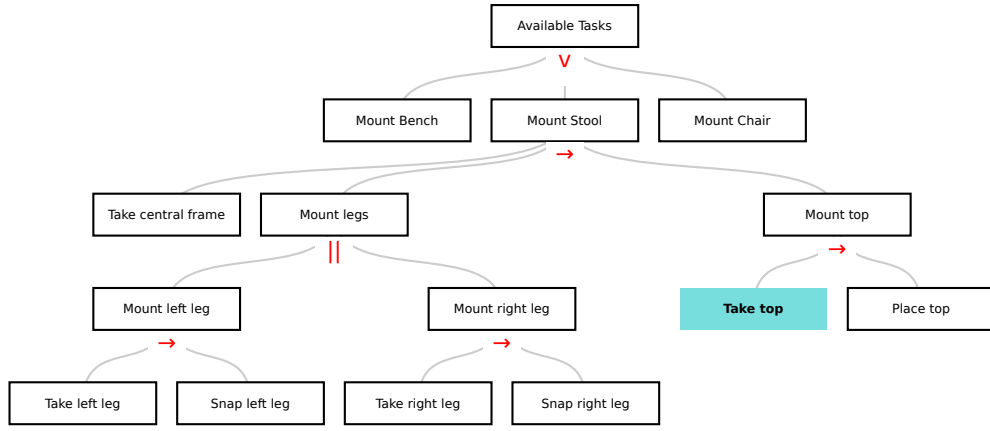


Fig. 2. User interface for high-level task representation. The user can inspect the task at various levels of granularity, and get a feedback on the subtask the robot has been instructed to accomplish (\rightarrow for sequential tasks, $||$ for parallel tasks, \vee for alternative tasks). During task execution, the robot can provide feedback to the user by highlighting its estimate of the current subtask (cyan block in picture). The task depicted is the joint construction of a stool (cf. Section IV-B).

to explicitly model decisions about communication acts; a question that [19] also targets from the different perspective of optimizing the formulation of a request for help.

A. Approach

We target robotic systems that are readily available to non-expert users. In this respect, a hierarchical representation of the task serves as the entry point for such users: it models the task with the same level of abstraction that one would use to describe the task to another person, while hiding the details of the implementation. An example of such model is shown in Fig. 2: each of the nodes of the graph are either a subtask or an atomic action that can be executed by either of the partners (or both). Without loss of generality, we extend the CC-HTM representation presented in [4] by introducing a new topological element in the task representation, that is the *alternative* operator (\vee in Fig. 2). This operator, which joins the already available *sequential* and *parallel* (respectively \rightarrow and $||$ in Fig. 2) operators, allows for a disjunction between subtasks, and adds a significant amount of expressiveness to the network introduced in [4].

As mentioned in Section I, the first contribution of this work is a system able to convert high-level hierarchical models into low-level task planners capable of being executed by the robot. We propose to transform high-level HTMs into low-level, on-line planners by using POMDPs, that model the commitment of robot and human to parts of the task and hence address the task allocation problem. POMDPs are also able to explicitly plan for actions that do not affect the state of the system but whose role is to reduce uncertainty about the system. This property enables communicative actions to be treated as first-class actions available to the system’s policy (cf. Section III-B). Furthermore, we consider here the extreme situation of a *blind robot* that is not capable of observing the actions of the human. This forces us to investigate mechanisms of coordination through communication only.

Importantly, a robot being able to plan the task at a high level of abstraction serves as the first step toward a

transparent, natural interaction with the human. An efficient, collaborative robot needs to reason about the kinds of supportive behaviors it can provide given the state of the world and the human collaborator, and should be able to eventually steer the human partner in order to optimize the task according to some metric. This degree of proactiveness enables a larger scope of socially-aware and natural collaborative systems. Firstly, it is possible to reason on regular planning problems, e.g. ‘Should the human or the robot take care of a specific action?’, the answer of which depends some metric to optimize. Secondly, the system can also reason about unobserved transitions: for example, if the human chooses a specific execution plan when multiple are allowed, the robot is not able to observe the human intention, and may gather information with a question. Optimizing under uncertainty enables taking such action only when beneficial, that is only if it is expected to impact the outcome of the robot’s decisions. This situation also arises when the robot does not observe directly whether the human has finished some subtask (as with fine manipulation tasks). Analogously, to limit uncertainty on the human side, the robot should make its own decisions transparent to the user. Such communication in turn reduces uncertainty on the future actions from the human. Finally, considering the cost of various actions, the robot can decide which communication channel to use. In particular, when making an error is not too costly, it may be preferable to try a physical action in an uncertain context rather than engaging in a complex communication process.

B. Restricted Model (RM)

We propose an automated technique able to transform task-level HTMs into low-level POMDPs. We focus on the optimal exploitation of communication as a (costly) way to reduce uncertainty on the system without affecting its state. To fulfill this goal, we convert each primitive subtask (that is, each leaf composing the HTM in Fig. 2) into a small, modular POMDP, which we call a restricted model

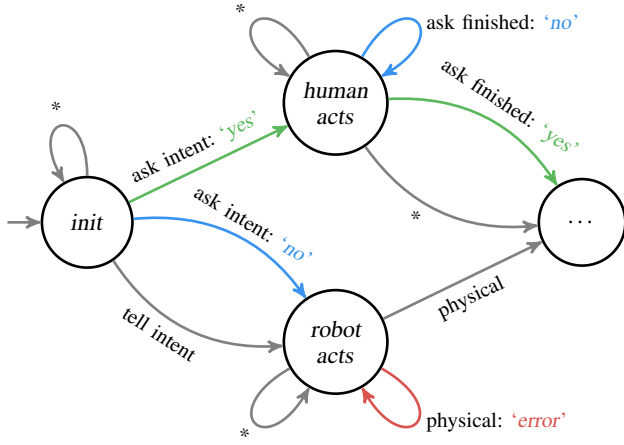


Fig. 3. Each subtask is modeled as an atomic POMDP. Nodes represent states, whereas links represent the [action, observation] pair that connects states together. For the sake of clarity, only the most relevant actions for each state are depicted, even though the planner is allowed to choose any action in any state. See Section III-B for details.

(RM [20]). Hence, the RMs are mostly independent from the rest of the problem and can be studied in isolation. Differently from [20], each RM is then composed at a later stage and the problem is solved in its entirety. This approach benefits from the modularity of the HTM representation, without the typical sub-optimality of policies that do not consider the full problem.

Fig. 3 depicts the restricted model developed in this work. Each leaf—that is, each atomic action in the task—can be ideally assigned to either the human or the robot. The state space S of the restricted model is composed of three states. The initial state (*‘init’* in Figure) is used to negotiate role assignment in the restricted model. Here, the system takes care for this duty: according to some metric (here minimal overall completion time, see Section IV-B), the system decides whether to communicate that the action is going to be performed by the robot (*‘tell intent’* in Figure) or to suggest that the human partner should be allotted to the task (*‘ask intent’*, question to which the human may answer *‘yes’* or *‘no’*). In both cases a transition occurs to the corresponding state (*‘human acts’* or *‘robot acts’*), where either the human or the robot is assigned and publicly committed to the task. In the current version of our system, both peers are required to follow through with the action once roles are assigned in the negotiation step. That is, we do not allow the agents to change their plan mid-action. Then the agent (human or robot) needs to execute the expected *‘physical’* action to trigger the transition to the first *‘init’* state of the next (or one of the next) subtask. This mechanism iterates for all the leaves composing the HTM representation until an end state is reached and the task is completed.

The pairs [action, observation] link the various states in the graph. In this work we restrict to observations that represent the way in which the human partner can interact with the system; possible observations are: *‘yes’* (green), *‘no’* (blue), *‘error’* (red). In addition, the *‘nothing’* observation (where the human does not reply or is not needed to) is also

taken into account (gray link in Figure). Five actions can be taken by the robot: *‘ask intent’* and *‘ask finished’*, to gather information about the human state, yield a *‘yes’* or *‘no’* observation; *‘tell intent’*, to communicate the robot’s intentions to the human partner, does not require an answer (*‘nothing’* observed); *‘physical’*, to actually complete the subtask (*‘nothing’* or eventually an *‘error’* in case of failure are observed); and finally, *‘wait’* always yield the *‘nothing’* observation. Importantly, the system is allowed to pick any irrelevant action in any possible state (*‘*’* in Figure), but will receive a *‘nothing’* or *‘error’* observation.

This describes the basics of the transition probabilities T and observation probabilities O of the restricted POMDP; however they include more stochasticity as described below. We allow, with a small probability, the robot actions to lead to an error that does not change the current state, or the human to skip answer to a question from the robot. This forces the solver to plan for unexpected or unobserved events and favors more robust policies. Similarly, while the human is committed to achieving the current subtask (*‘human acts’* state), the model accounts for the durations of the robot’s actions by varying the probability that the human completes his or her action (i.e. the probability of transition to the next state) depending on the duration of the robot’s action (such as *‘wait’* or *‘ask finished’*). Specifically, we model the completion of the human action as an exponential decay whose mean time is the average duration of the action. Given a robot action, the event that the human finishes during this action is modeled as a Bernoulli variable; its probability corresponds to the probability of the decay during the duration of the robot’s action.

C. Assembling the modular POMDP

From the HTM we derive the list of successors for each node and compute the total number of states and actions of the global POMDP. We then populate the matrices representing transitions, observation, and reward probabilities to encode the RM from Fig. 3, for each node as well as for node to node transitions. The topological operators introduced in Section III are accounted for when restricted models are linked together. For what concerns a sequential operator (\rightarrow in Fig. 2), each *‘init’* state represents the final state of the previous model. Alternative operators (\vee) branch out the POMDP in multiple paths with equal transition probability—the choice of the path will be dictated by their intrinsic costs or randomly in case of equal costs. Finally, it is always possible to convert a parallel operator (\parallel) into an alternative between paths containing the subtasks the operator is composed of, but in a differently ordered sequence.

Scalability is a common issue when POMDPs are applied to problems with a large number of states. In our approach, since we consider a high level representation of the task and each subtask introduces three new states, the size of the state space increases linearly with the number of leaves in the HTM. In the realistic scenario we present, this system is used in contexts in which an upper bound to the number of leaves is the number of subtasks that an human operator can

physically handle. Hence keeping the state space relatively small. In extreme cases where the state space would be too big, our framework intrinsically accounts for the ability to split the full problem into smaller POMDPs and retrieve a successful (although sub-optimal) policy—similarly to [20]. We however do not implement this strategy in this paper.

IV. IMPLEMENTATION

A. Experimental Setup

We implemented our system on a Baxter Research Robot (cf. Fig. 1), using the Robot Operating System (ROS, [21]). As mentioned in Section III, although we did not place our focus on boosting its overall skills, we devised a set of basic capabilities in order for the robot to be an effective partner. To this end, we implemented two low-level, stateless controllers (one for each of the robot’s arms) able to operate in parallel and to communicate one another if needed. A library of high-level predefined actions (in the form of ROS services) is available for the POMDP planner to choose from; such actions range from the simple, single arm ‘*pick object*’ to the more complex ‘*hold object*’ (which requires a physical collaboration with the human partner) or ‘*hand over*’ (which demands a bi-manual interaction between the two arms). Inverse kinematics is provided by *TRAC IK* [22], an efficient, general-purpose library that has been adapted to the Baxter robot and guarantees a control rate of 100Hz. For both controllers, the perception system is based on Aruco [23], an OpenCV-based library able to generate and detect fiducial markers. Each object in the workspace is equipped with a unique ID associated with a specific marker; by exploiting the robot’s kinematic model and receiving visual feedback from the cameras placed on the end effectors, the robot is able to track objects into its three-dimensional operational space. This is particularly useful in order to achieve precise localization and grasping skills, because it is possible to implement visual servoing algorithms with a feedback loop that is as close as possible to the interaction itself.

The system allows for context-based, multi-modal interactions with the user. Multiple, redundant communication channels are exposed by the framework, with the goal of interacting with the user on ‘*human terms*’ [24]. This redundancy aims at allowing the human partner to choose the interface that suits him the most; importantly, it also has the advantage of making the interaction itself more robust. The exposed layers are the following:

1) *Web Interface*: it enables bi-directional communication between the robot and the human peer. The user can load new tasks, inspect the current task, and receive feedback from the robot’s understanding of the state of the task (see Fig. 2 for details). It is based on *roslibjs*, a JavaScript library that interfaces with ROS [25].

2) *Feedback Channel*: it is presented to the user in the Baxter’s head display (see Fig. 1). It allows the user to receive constant feedback about the robot’s state and intents.

3) *Text-to-Speech (TTS) System*: based on the SVOX-PICO engine¹, is used to verbally interact with the human peer during task execution. For the purposes of this work, the verbal interaction is limited to yes/no questions, that can be answered by the user in the web interface. The questions are also prompted in the Baxter’s head display.

4) *Force Interaction Layer*: it endows the robot with the capability of detecting when a specific force pattern is applied by the human partner to the robot’s arm. This is particularly important to accomplish natural interactions during physical collaborative actions (for example holding a lumber while the human drills into it).

5) *Emergency Channel*: it allows the human partner to send error messages to the controllers: it can be triggered by either pressing one of the buttons on the robot’s end-effector, or by pressing an appropriate button on the web interface, if the arm is not within reach.

B. Experiment Design

We devised an experiment in which the human and the robot are engaged in the joint construction of flat-pack (IKEA-like) furniture—specifically a stool (cf. Fig. 1). It is a miniaturized version of a real stool commercially sold online², and is composed of four parts (a central frame, the right and left legs, and the top part) that simply require being snapped together—a task that the Baxter is not able to perform in its entirety. The constituent parts are placed in a pool of objects that can be accessed by the robot exclusively (see Fig. 1). The human and the robot are allowed to interactively build the stool in a common workbench. It is worth noting how, although simple, this experiment represents an ideal scenario to evaluate the proposed approach: we purposely chose a task that does not require any particular skill from the human partner, but where the roles of both the human and the robot are decidedly clear. That is, there is no need for the experimenter to clarify to the human the role assignment in this task: each of the four parts that constitute the stool require a retrieval from the pool of parts (to be performed by the robot), and eventually a snap onto the frame (to be performed by the user).

The metric used to assess the effectiveness of our approach is the *completion time* that the overall human-robot collaborative system takes to complete the task. As a control condition, we compare our technique with a similar system where no anticipation and task reasoning is in place. That is, a system in which the human operator actively asks for specific actions to be performed by the robot through a web interface. The skill set and capabilities of the robot are the same in both scenarios. In the following paragraphs, the two experimental scenarios are presented. For both the experiments, completion time between the beginning and the end of the full task are recorded, as well as completion time for each of the subtasks the stool assembly is composed of.

¹ <https://en.wikipedia.org/wiki/SVOX>

² The CAD design of the stool has been downloaded from opendesk.cc, and then 3D-printed with a 1:2 scale.

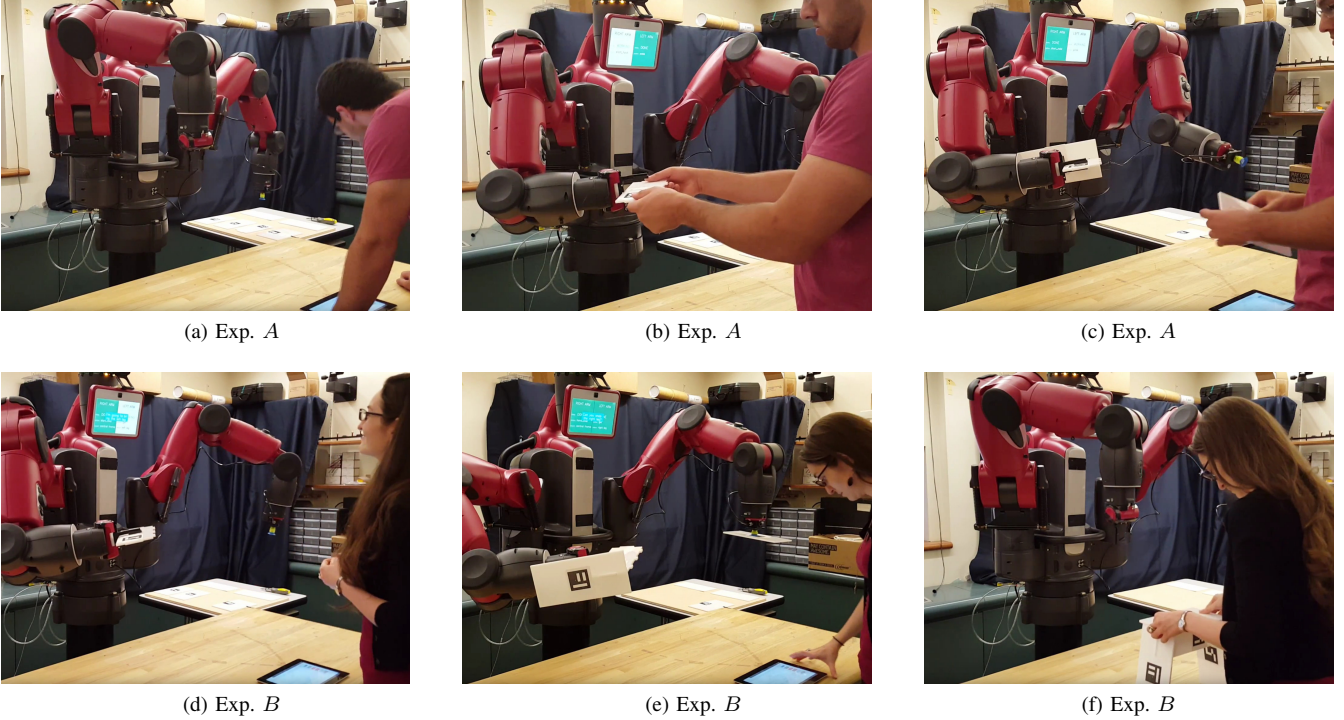


Fig. 4. Snapshots acquired during the collaborative assembly of the stool in the control and the interactive conditions. Control condition (*top*): a) The participant asks for the central frame through the web interface (tablet on table); b) the robot supports the human by holding the central frame in place; c) the user snaps the right leg onto the central frame that is held by the robot. Interactive condition (*bottom*): d) the robot communicates the intention to get the central frame (as computed by the POMDP); e) the robot asks the human to snap the right leg onto the central frame, because the human is deemed more efficient (third column in Table I); f) after the last subtask is completed, the POMDP enters a final (virtual) state.

1) *Exp. A - control condition*: in this scenario, the human is required to perform the task in conjunction with the robot by explicitly requesting robot actions when needed (similarly to the Explicit Teaming Group in [7]). Firstly, the human participant is introduced to the Baxter and the task. The experimenter shows the capabilities of the robot (that are the ability to retrieve an object from the pool and to hold a part in place when supporting the human), and briefly illustrates how to complete the stool task. Finally, the human is presented with the web interface (cf. Section IV-A.1), which displays HTM for the task under completion (cf. Fig. 2), and a series of five buttons that trigger the robot actions (that are ‘get central frame’, ‘get left leg’, ‘get right leg’, ‘get top’, ‘hold part’, respectively). The human participant is allowed to complete the task without any restrictions.

2) *Exp. B - interactive condition (collaborative assembly with role assignment and task allocation)*: this scenario introduces a shift in task allocation responsibilities from the human to the robot. That is, the human is freed from reasoning about the task and what action each partner is going to perform. In this case, the experimenter only anticipates to the participant that the robot is going to be more proactive and communicative. The web interface is now simplified, and the action buttons are replaced with a ‘yes’ and ‘no’ button (to reply to the robot’s questions), plus the ‘error’ button that the human is instructed to press if the robot is taking the wrong action or is generally in error. Importantly, these

buttons explicitly correspond to the type of observations that the POMDP planner allows for, and they represent the on-line interaction channel that the human partner is provided with in order to affect the execution of the task (see Section III-B).

V. RESULTS

The framework has been released under the LGPLv2.1 open-source license, and it is freely accessible on GitHub³. Although the majority of the codebase is robot-independent, the control architecture is readily available for any Baxter robot. We test the proposed system in a proof-of-concept scenario in which the human and the robot are engaged in the joint construction of a stool. We define the problem as a sequence of 9 subtasks; this translates to a state space dimensionality of 28 (9×3 state for each restricted model, plus 1 final state), and an action space composed of 37 possible actions (9×4 , where 4 is the number of possible actions described in Section III-B, plus a ‘wait’ action).

The high-level actions presented to the user in the hierarchical task model fall into four different categories. As described in Section III-B, leaves in the HTM are then transformed into a restricted model with equal treatment. To showcase the flexibility of the proposed approach, we enforced different costs for each action typology in order for the POMDP solver to act differently in each case. Table I

³github.com/scazlab/baxter-collaboration hosts the source code for the Baxter’s controllers, whereas github.com/scazlab/task-models hosts the HTM to POMDP planner.

TABLE I
COSTS (COMPLETION TIMES) FOR THE STOOL TASK.

	<i>Get</i>	<i>Hold</i>	<i>Snap</i>	<i>Release</i>	<i>Communicate</i>
<i>Human Acts</i> [s]	∞	5	5	∞	N/A
<i>Robot Acts</i> [s]	20	10	∞	1	2
<i>Robot Error</i> [s]	10	∞	∞	1	N/A

illustrates the costs used to allocate a specific subtask to either the human or the robot, in terms of realistic completion times estimated from prior experimental sessions. For each high-level action, the completion times in seconds for the human and the robot are shown (‘*Human Acts*’ and ‘*Robot Acts*’ respectively). Additionally, the POMDP solver allows for a cost to be paid if the robot performs a ‘*physical*’ action in the wrong state (‘*Robot Error*’ in Table). As described in Section III-B, each action can be theoretically performed by both peers, but the stool task has been explicitly designed to constrain some actions to either the robot or the human exclusively, in order to enforce role assignment (cf. Section IV-B). To this end, we assign an infinite completion time for the peer that is not supposed to perform the task. Additionally, a constant cost of 2 seconds has been designated for each communicative action. The system has been evaluated with 8 participants, in a within-subjects design. All experimental conditions are to be performed twice for each participant, for a total of four demonstrations—hereinafter referred to as *A1*, *A2*, *B1*, and *B2*. To reduce the chance that familiarization with the task would penalize the first demonstrations, we randomized the initial condition—i.e. four participants started their experiment with condition *A*, whereas the other four started with condition *B*. Nonetheless, some degree of improvement in terms of completion time between the first and the second demonstration pertaining to a specific condition (i.e. *A1* vs. *A2* and *B1* vs. *B2*) is still expected. Additionally, the two experimental conditions were alternated—i.e. participants were allowed to perform experiments following two trajectories: either [*A1*-*B1*-*A2*-*B2*] or [*B1*-*A1*-*B2*-*A2*].

To solve the POMDP problem, we employ the solver developed in [26]. For all the participants, each POMDP solution took less than 10 seconds to compute: that is, the POMDP solver took less than 10 seconds to solve the full task. A series of snapshots illustrating the task is shown in Fig. 4. Please refer to the accompanying video for a more extensive demonstration of the experimental session (full resolution available at youtu.be/l6alHuMqx6Y).

Fig. 5 depicts the completion time that each participant needed to complete the four demonstrations. A two-samples *t*-test between conditions *A* and *B*—see Table II for a comparison of the average completion time and standard deviation among demonstrations—shows statistically significant difference, with a *p*-value of 0.016. Overall, the results suggest evidence of an improvement in terms of task efficiency: the total time spent in carrying out the task was

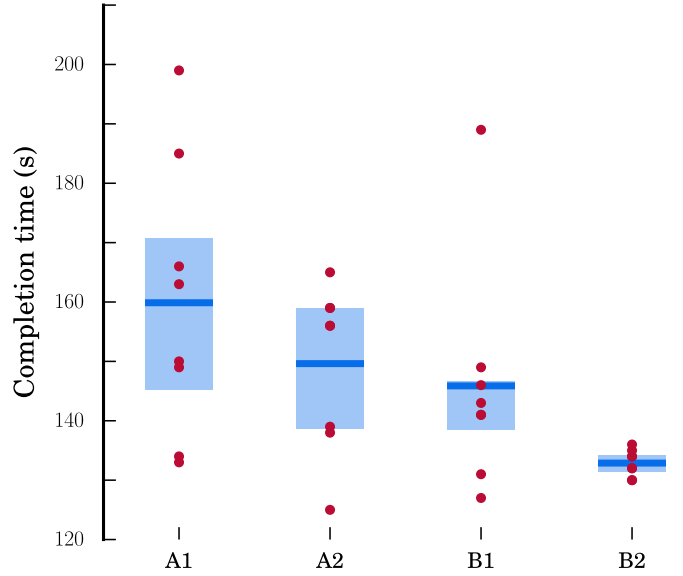


Fig. 5. Completion time [s] for each participant involved in the construction task. A total of 8 participants (red dots in figure) performed two baseline conditions (*A1* and *A2*), and two interactive sessions (*B1* and *B2*). For each trial, average completion time and standard deviation are shown in blue.

TABLE II
AVERAGE COMPLETION TIME [S] DURING THE CONTROL (EXPERIMENT A) AND THE INTERACTIVE (EXPERIMENT B) CONDITIONS.

Exp A - Control 154.8 (21.8)		Exp B - Interactive 139.4 (14.7)	
<i>A1</i>	<i>A2</i>	<i>B1</i>	<i>B2</i>
159.9 (21.8)	149.6 (13.0)	145.9 (17.7)	132.9 (2.1)

reduced by 10%. Interestingly, the condition *B2*—that is, the second demonstration of the interactive condition—is considerably better in terms of standard deviation. This could be interpreted by the fact that all the participants were hitting a plateau for what concerns how fast they could complete the task, since in such a scenario more responsibility is allotted to the robot, and less variability is allowed to the human in terms of number of interactions with the system.

We have registered a general user preference toward the proposed system. Interestingly, multiple reasons were given that support the idea that a more transparent interaction favors the collaboration between peers. Most of the comments suggested a reduced cognitive load as the reason for such preference. A non-exhaustive list is the following: i) there is no need to exactly remember the steps needed to complete the task, or to review such steps in the web interface; ii) there is less time ‘wasted’ in taking decisions (i.e. the system ‘helps’ the user gain time); iii) the web interface is simpler and less cluttered, and the interaction with it is less frequent—as a matter of fact, the number of buttons was reduced from 8 to 3. These comments are of particular interest if put in perspective: as detailed above, we considered a simple assembly task with a limited number of

steps; we foresee that the impact of the proposed system will be even more substantial in real-world assembly tasks, when the number of steps is considerably higher.

VI. DISCUSSION AND CONCLUSIONS

In this work, we demonstrate how high level task models can be combined with adaptive planning under uncertainty, as modeled by POMDPs. We propose a framework able to autonomously reason about the problem of allocating specific subtasks to either the robot or its human peer. That is, the robot intercedes for the human during the decision making step. Importantly, the system is completely transparent to the user, who is in full control of both the decisions made by the robot, and its operations during execution. We present a pilot experiment where such system is shown to be significantly more efficient than a comparable scenario where the decision making role is allotted to the human.

It is worth noting that in all the conditions the user was provided with minimal information regarding the capabilities of the framework. Indeed, the ability of the proposed system to intuitively interact with the human partner is considered an important asset of the proposed approach.

A technical limitation of the system is arguably the fact that the decay model for the completion of human actions might not be suited for all kinds of human actions. Indeed modeling long actions with predictable durations would require some memory of the remaining time to completion. However evaluating which actions deviate from the model, how this may affect the computed robot behavior, and how modifying the model could alleviate such issues, is a matter for future work.

Further work has to formally demonstrate the full scalability of the proposed work to more complex tasks, involving more complex actions. In particular, efficient solvers might have to be developed to alleviate the exponential growth of the problem size, by e.g. leveraging the relative sparsity and block structure of the transition matrix. Additionally, extensions of the current model to multi-agent planning and parallel execution of human and robot actions may turn essential for real-life efficient applications. Similarly, more in-depth user studies will be able to evaluate the diversity and quality of the behaviors generated by the system in more complex scenario. The influence of the many system parameters on real user experience is still unknown. Other aspects such as finer modeling of the influence between the human and robot on decisions could improve the framework.

REFERENCES

- [1] L. Johannsmeier and S. Haddadin, “A Hierarchical Human-Robot Interaction-Planning Framework for Task Allocation in Collaborative Industrial Assembly Processes,” *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41–48, Jan 2017.
- [2] M. Toussaint, T. Munzer, Y. Mollard, L. Y. Wu, N. A. Vien, and M. Lopes, “Relational activity processes for modeling concurrent cooperation,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5505–5511.
- [3] A. Garland and N. Lesh, “Learning hierarchical task models by demonstration,” *Mitsubishi Electric Research Laboratory*, 2003.
- [4] B. Hayes and B. Scassellati, “Autonomously Constructing Hierarchical Task Networks for Planning and Human-Robot Collaboration,” in *International Conference on Robotics and Automation (ICRA)*, 2016.
- [5] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, “IkeaBot: An autonomous multi-robot coordinated furniture assembly system,” in *IEEE International Conference on Robotics and Automation*, 2013.
- [6] J. Shah and C. Breazeal, “An empirical analysis of team coordination behaviors and action planning with application to human-robot teaming,” *The Journal of the Human Factors and Ergonomics Society*, vol. 52, no. 2, pp. 234–245, 2010.
- [7] J. Shah, J. Wiken, B. Williams, and C. Breazeal, “Improved human-robot team performance using chaski, a human-inspired plan execution system,” in *Proceedings of the 6th international conference on Human-robot interaction*. ACM, 2011, pp. 29–36.
- [8] K. Erol, J. Hendler, and D. S. Nau, “HTN Planning: Complexity and Expressivity,” in *In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*. AAAI Press, 1994, pp. 1123–1128.
- [9] R. A. Knepper, D. Ahuja, G. Lalonde, and D. Rus, “Distributed assembly with and/or graphs,” in *Workshop on AI Robotics at the Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.
- [10] K. P. Hawkins, S. Bansal, N. N. Vo, and A. F. Bobick, “Anticipating human actions for collaboration in the presence of task and sensor uncertainty,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2215–2222.
- [11] N. Mehta, S. Ray, P. Tadepalli, and T. Dietterich, “Automatic discovery and transfer of MAXQ hierarchies,” in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 648–655.
- [12] J. Hoey, P. Poupart, A. von Bertoldi, T. Craig, C. Boutilier, and A. Mihailidis, “Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process,” *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 503–519, 2010.
- [13] S. Nikolaidis, P. Lasota, R. Ramakrishnan, and J. Shah, “Improved human-robot team performance through cross-training, an approach inspired by human team training practices,” *The International Journal of Robotics Research*, vol. 34, no. 14, pp. 1711–1730, 2015.
- [14] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, “Information gathering actions over human internal state,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ Int. Conf. on*, 2016, pp. 66–73.
- [15] M. P. Woodward and R. J. Wood, “Framing Human-Robot Task Communication as a POMDP,” *CoRR*, vol. abs/1204.0280, 2012.
- [16] N. Gopalan and S. Tellex, “Modeling and Solving Human-Robot Collaborative Tasks Using POMDPs,” in *RSS Workshop on Model Learning for Human-Robot Communication*, 2015.
- [17] B. Hayes and B. Scassellati, “Discovering task constraints through observation and active learning,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 4442–4449.
- [18] C. Breazeal, G. Hoffman, and A. Lockerd, “Teaching and working with robots as a collaboration,” in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*. IEEE Computer Society, 2004, pp. 1030–1037.
- [19] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy, “Asking for help using inverse semantics,” in *Robotics: Science and Systems*, 2014.
- [20] G. Shani, “Task-Based Decomposition of Factored POMDPs,” *IEEE Transactions on Cybernetics*, vol. 44, no. 2, pp. 208–216, Feb 2014.
- [21] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA Workshop on Open Source Software*, 2009.
- [22] P. Beeson and B. Ames, “TRAC-IK: An Open-Source Library for Improved Solving of Generic Inverse Kinematics,” in *Proceedings of the IEEE RAS Humanoids Conference*, Seoul, Korea, November 2015.
- [23] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014.
- [24] C. Breazeal, *Designing Sociable Robots*. MIT Press, 2002.
- [25] R. Toris, J. Kammerl, D. V. Lu, J. Lee, O. C. Jenkins, S. Osentoski, M. Wills, and S. Chernova, “Robot Web Tools: Efficient messaging for cloud robotics,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 4530–4537.
- [26] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial intelligence*, vol. 101, no. 1, pp. 99–134, 1998.