An Affordance-based Action Planner for On-line and Concurrent Human-Robot Collaborative Assembly

Francesca Stramandinoli[1], Alessandro Roncone[2], Olivier Mangin[2], Francesco Nori[1] and Brian Scassellati[2]

***Abstract*— We propose an affordance-based action planner for on-line and concurrent human-robot collaboration on a shared task. The proposed planner enables the robot to: (i) derive a high-level manipulation strategy of a joint task that requires the performance of a sequence of actions of both the robot and the human and (ii) decide when to intervene by taking action into the sequence of actions performed by the human. The robot intervention in the course of action is dictated by the anticipation of the needs of the human co-worker; the robot can proactively perform a supportive behavior to help its human partner. For building plans shared between the robot and the human, we exploit the knowledge represented by affordance models. Affordances are leveraged to tailor the plan to the environment where the robot operates, selecting the best action to implement a step of the plan according to the object features and the human preferences. The proposed planner has two important features: (i) reaction to action failure to dynamically adapt the plan during the execution, and (ii) planning of concurrent actions that increases the level of support that the robot can provide, and improves the working conditions of the human.**

We applied the proposed affordance-based planner to an assembly task for the demonstration of on-line and concurrent collaboration using the Baxter robotics platform.

I. INTRODUCTION

Due to safety reasons, industrial robots traditionally operate in cages away from human factory workers. Recent advancements in robotic technology and safety mechanisms have led to a new generation of industrial robots conceived for collaborating with human workers in manufacturing tasks that cannot be fully automated (e.g. manipulating objects that are deformable). In some cases, semi-automation is preferable to full automation. Indeed, the combination of industrial robot capabilities (e.g. perform tasks in an accurate, precise and fast way) with human perceptual, motor and cognitive skills can increase efficiency, quality and productivity. Human workers have knowledge about the tasks to perform and they can think of more efficient ways to organize the work. Moreover, the collaboration of human workers with their robotic counterpart allows a flexible organization of the tasks to be executed and opens to the possibility of introducing improvements in the way the tasks are executed. However, in building robotic systems

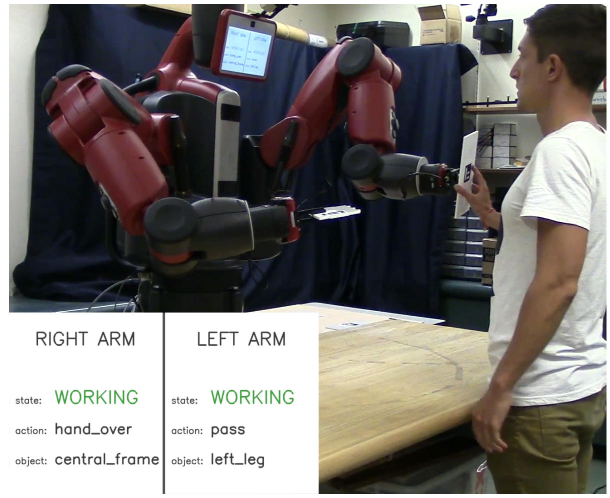


Fig. 1: The Baxter robot and a human worker engaged in a collaborative assembly task. The robot can use both arms to concurrently support the human partner: (i) it can hold objects to facilitate the human in the assembly of an object, (ii) it can pass objects that are out-of-reach for the human partner. Please refer to the accompanying video for a demonstration of the human-robot collaboration on the assembly of a stool (full resolution available at `https://youtu.be/d1A-kxW1Rsw`).

that can perform supportive behaviors [1], several challenges need to be addressed (e.g. dexterous manipulation, planning algorithms, social behaviors, safety mechanisms, etc.).

One of the first works proposed on building robots that can collaborate with humans in shared tasks has argued that speech, gesture and expressive cues, can be used to coordinate and synchronize the behavior of the robot and the human during the collaboration [2]. Most importantly, the study presented in [2] has suggested that during human-robot collaboration on a joint task, the robot needs to understand the person's intentions in order to behave as a partner rather than just a tool.

We propose an affordance-based action planner for the on-line and concurrent human-robot collaboration on a shared task (Fig. 1). Our approach enables the robot to (i) predict the behavior of the human agent depending on the environment in which they collaborate, and (ii) determine the action that the robot shall perform to provide support to the human partner. We propose to exploit the concept of affordance, which was first introduced by J. J. Gibson [3] for referring to all the motor programs that an acting organism can perform during

[1]Francesca Stramandinoli and Francesco Nori are with the iCub Facility Department, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genoa, Italy (email: {francesca.stramandinoli, francesco.nori}@iit.it)

[2]Alessandro Roncone, Olivier Mangin and Brian Scassellati are with the Social Robotics Lab, Computer Science Department, Yale University, 51 Prospect St, New Haven, CT 06511-8937, US (email: {alessandro.roncone, olivier.mangin, brian.scassellati}@yale.edu)

an interaction with a specific object in the environment. We model affordances through Bayesian Networks (BNs) [4] that capture relations between (A)ctions, (O)bjects, and (E)ffects. In such a framework, the planner leverages the affordance model to tailor generic (environment-independent) operators stating effects to be achieved to specific actions that can be actually executed in the environment in which the robot operates. The probabilistic relation among actions, objects and effects is built through the experience of the robot during the interaction with the human, and can hence reflect the preference of the human partner(s) in the behavior of the robot. To achieve a certain effect, the robot will select the behavior that has statistically received the most positive feedback from the human partner in the same environment conditions. This results in an affordance-based action planner that enables a robot to predict and anticipate the behaviors of its human partner, and to adapt its behavior to the human's preferences. The primary contributions of this work are:

- A framework that enables the concurrent and on-line participation of a robot in a shared task with the human partner.
- Evaluations showcasing the application of the affordance-based planner to an assembly task with the Baxter robotics platform.

Previous studies that have investigated the use of affordances for multi-step action planning, with the exception of the study presented in [5], did not deal with the learning of shared plans between a robot and a human. Differently, our framework targets supportive behaviors, and has a unique architecture that integrates a planner based on Hierarchical Task Networks (HTNs) [6] and affordance models based on BNs. HTNs provide the capability to hierarchically decompose complex tasks, with a positive impact on the scalability of the framework. BNs can adapt the plan execution to user preferences, by learning the probabilistic relations among the entities of the affordance models through repeated interactions of the robot with human agents.

II. RELATED WORK

An extensive amount of work has been carried out in the area of motion and task planning [7]. Recently, a new optimization-based approach has been applied to sequential robot manipulation [8]. Among other possible approaches proposed for dealing with planning problems, hierarchical planning has been a rich research domain. For example, a domain-independent planning system (SHOP2) based on Hierarchical Task Networks has been presented in [9]; SHOP2 generates the steps of each plan in the same order that those steps will be executed. A hierarchical approach for interleaving planning and execution has been suggested in [10]. Along the same line of research the Hierarchy Planning in the Now (HPN) has been proposed for the execution of geometrical problems [11]. A new type of Hierarchical Task Network (Clique/Chain Hierarchical Task Networks, CC-HTNs), alongside an algorithm for autonomously constructing them from topological properties derived from graphical task representations, have been presented in [12].

Several approaches have been developed for reducing search complexity in planning (i.e abstractions, heuristics, etc.). Recently, affordance-based action abstraction has been proposed in robot task planning [13]; following the approach in [13] an agent learns which actions can be substituted and executed successfully in a given context (e.g. when a robot has to `move` an object, it predicts the behavior that can be substituted for `move` in order to achieve its goal (e.g. pick and place, push, pull, etc.)).

The robotics community has devoted significant attention to multi-agent teamwork. For example, a framework for multi-agent team coordination in robot soccer has been presented in [14]. In [15] an automated assembly system that directs the actions of a team of heterogeneous robots in the completion of an assembly task has been proposed. A novel task and motion planning approach that implements the generation of supportive behaviors enabling a teammate to reduce cognitive and kinematic burdens during task completion has been presented in [1].

III. SYSTEM ARCHITECTURE

We describe a framework that enables the collaboration of a human with a robot on a shared task. The human-robot collaboration is achieved through the development of different capabilities that allow: (i) the decomposition of a task in elementary actions and (ii) the determination of the user preferences in the human-robot cooperation. Our framework leverages an affordance-based Hierarchical Task Network (HTN) planner. The HTN provides the hierarchical decomposition of non-primitive tasks into a list of sub-tasks until primitive tasks are reached [6]. We assume that the robot has access to such a decomposition that may have been learned from demonstration [16] or defined by a human partner. Each primitive task is grounded in affordances that capture the relations between the actions that an agent can perform on objects to obtain desired effects [3]. Affordances, modeled as Bayesian Networks, capture the robot knowledge of the environment, and are leveraged for two purposes:

- **Predict the behavior of the human agent**. BNs are used to correlate the behavior of the human agent with environmental conditions (e.g. object's properties). The robot can hence determine whether the human agent will execute an action himself or will need support.
- **Determine the action that the robot should perform to provide support to the human partner**. BNs are used to capture the preference of the human agent in the execution of an action. Each step of the plan can be executed in different ways; the robot will learn from the observation of the human agent the preferred execution and perform the task accordingly.

Fig. 2 shows the architecture of the proposed framework. A human agent interacts with the robot to communicate the task to perform. This information is passed from the robot to the planner for determining the steps for the execution of the task. The planner uses the methods and operators of the HTN planner to decompose the task (cf. Fig. 2), and queries the robot knowledge model to determine (i) which steps

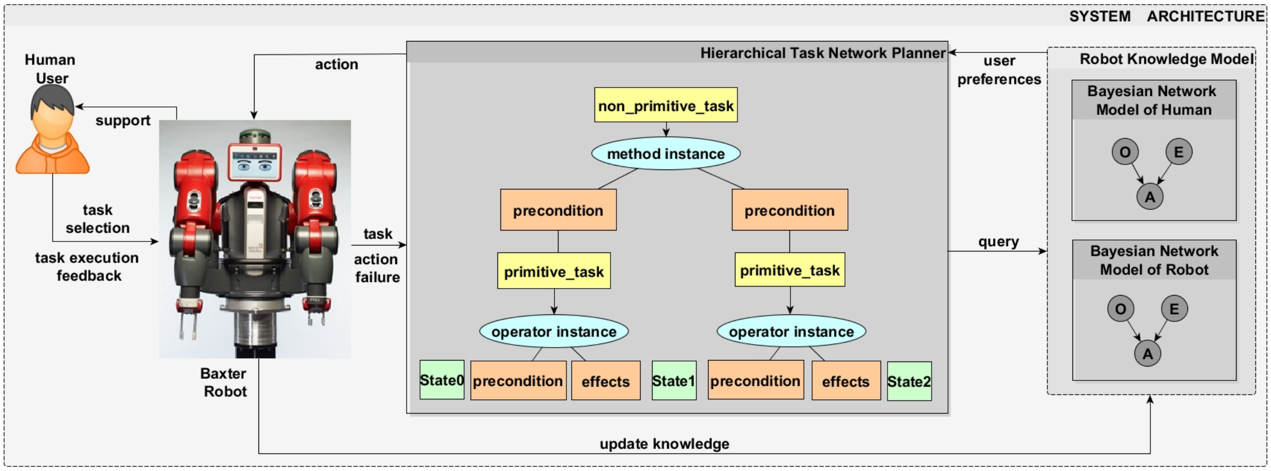


Fig. 2: Illustration of the system architecture that enables the collaboration of a human with a robot on a shared task. The human-robot collaboration is achieved through the development of different capabilities that allow: (i) the decomposition of a task in elementary actions (HTN planner) and (ii) the determination of the user preferences (BNs models) in the human-robot cooperation.

do require support from the robot and (ii) how to perform an action. When the plan is ready, the planner controls its execution by the robot, according to the algorithm described in (Fig. 3). During plan execution, the robot provides support to the human, whereas the human can provide feedback to the robot to affect its operations. We identify two feedback scenarios:

- **Action failure**. In this case, the feedback is forwarded to the planner, and the planner re-evaluates the plan to recover from the action failure.
- **User preference**. The user communicates to the robot that the supportive behavior is not matching is expectations (e.g., he was expecting the robot to intervene/not intervene in an operation, or to perform a task using a different action). In this case, the feedback can be used for on-line training of the BN models, so as to better adapt future task execution to the user expectations.

In the following sections we describe in detail the constituent blocks of the system architecture depicted in Fig. 2 and their integration.

A. *Hierarchical Task Network Planner*

The HTN decomposes non-primitive tasks into a list of sub-tasks until primitive tasks are reached. Methods are parameterized descriptions of a possible way to perform a non-primitive task by performing a collection of sub-tasks. Indeed, there may be more than one method for the same task. Operators are parameterized descriptions of what the basic actions do; they can be executed directly. Similarly to STRIPS-like planners, operators can be applied to primitive tasks; upon the verification of pre-conditions (i.e. particular state of the world), they produce effects/post-conditions (i.e. state transitions). The effect of an action (i.e. post-condition) at time $t-1$ becomes the pre-condition for the action at time t, while the effect at time t is the post-condition of the action at time t. The constraint of verifying pre-conditions reduces the search space over which the planner searches the possible operators; in particular, given a certain state of the world, the only operators that will be executed are the ones that verify their pre-conditions.

Concurrent Execution and On-line Planning: Collaborative robots can perform multiple actions in parallel; as an example, they can use one arm to collect and pass parts to the human agent, while holding the object to be assembled with the other arm. Leveraging this capability increases the level of support that the robot can provide, and improves the working conditions of the human. In our framework, we exploit this capability assuming that each robot arm is an independent agent. The planner can send different actions to each agent, and the agents perform them in parallel. There are cases where the actions performed by the agents shall be coordinated (e.g., they need to be executed together). The coordination is managed by the planner, leveraging operator pre-conditions.

Our planner is capable to re-plan in case the execution of actions raises errors. The robot provides a feedback to the planner about the outcome of an action (i.e. success or fail), and the planner can track the progress in the task execution and eventually react to action failures. The failure is addressed by reiterating the failed operation; more complex re-planning strategies will be integrated in future evolution of the proposed framework. Nonetheless, in this work re-planning is successfully achieved while performing complex, multi-step actions, or while one action fails but the planner has already moved on the subsequent step.

The algorithm of the controller managing concurrent execution and on-line planning can be summarized as follows (Fig. 3):

1) **HTN Planner List**: The HTN planner determines the list of operators to execute for performing the task.
2) **Get Next Operator**: The planner picks the next operator to execute.

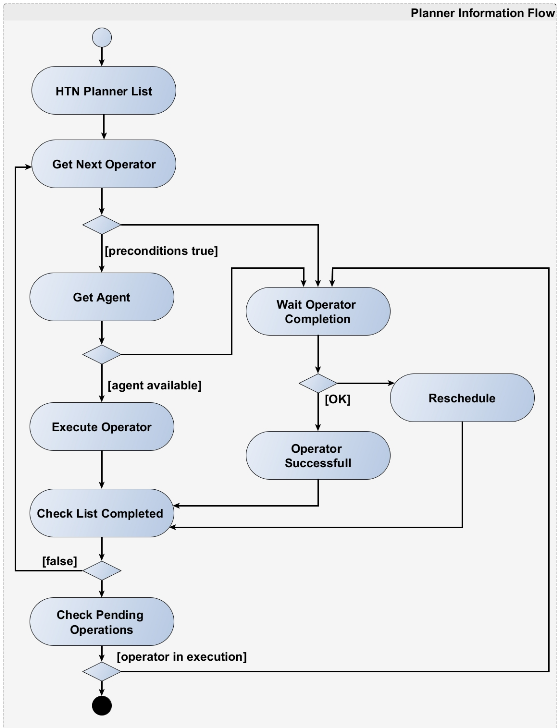


Fig. 3: Information flow in the controller managing concurrent execution and on-line planning.

3) **Get Agent**: Each operator is associated with a robot agent, determined by the HTN planner. If the operator preconditions are met, the planner determines the agent for the next operator.
4) **Execute Operator**: If the robot agent is free (it is not performing another operator), the agent starts the operator execution, and moves to the busy state.
5) **Wait Operator Completion**: If the operator preconditions are NOT met, or the robot agent is busy, the planner waits for the agent client to complete its activity. When an agent completes the execution of an operator, the planner re-evaluates the conditions to execute the next operator. Notice that the completion of an operator affects both the state of the agents (there is at least a free agent), but also (in case of successful completion) the state of the environment checked by the precondition of the next operator.
6) **Operator Successful**: When an operator has been successfully executed, the operator in the list is marked as successful: this allows tracking the progress in the plan execution.
7) **Reschedule**: In case an operator fails, the operator is rescheduled at the top of the operator list, as the next operator to execute.
8) **Check List Completed**: If there are still operators in the list, the planner moves to the next operator.
9) **Check Pending Operations**: If there are no more operators to process, but there are operations still in execution, the robot waits for them to complete. When all the operations have been completed, the task execution ends.

B. *Robot Knowledge Model*

The robot knowledge model is implemented by the BNs capturing the information about the human expectations on the robot behavior. The planner leverages this information to adapt the plan to the user and to the environment conditions (e.g., object position). The BNs share the same knowledge about the environment (i.e. object properties) and correlate them with the intended effect and the action to perform. The training of the BNs can be performed off-line, i.e., before the execution of the task, or on-line, while the task is executed. This allows a statistical characterization of the user preferences that in a multi-user scenario adapts the robot behavior to the most likely expectations of the user.

*1) **BN Model of the Human**:* this model allows the prediction of the behavior of the human partner. The model contains information on: (i) the object properties (e.g. object type, position, orientation), (ii) the desired outcome of the actions performed by the human and (iii) the intended behavior of the human. The latter can be: `Act`, stating that the human executes the action himself and `Support`, meaning that the human expects the robot to provide support, e.g. by moving the object closer to the human peer (the robot moves the object closer to the human, and the human places it). According to the environmental conditions (i.e. object's properties) the expected behavior is selected by the model.

*2) **BN Model of the Robot**:* when the robot has to provide support to the human co-worker, it determines the action to execute by using a BN model that contains information on: (i) the object properties (e.g. object type, position, orientation), (ii) the desired outcome of the actions performed by the human and (iii) the behavior that the robot has to perform for providing support to the human.

C. *Integration of the Planner with the Robot knowledge model*

The HTN planner and the Robot knowledge model are jointly utilized to establish a collaborative plan between the robot and the human agent (Fig. 4). Specifically:

- The HTN decomposes non-primitive tasks, recursively applying methods, into sub-tasks.
- The selection of methods can involve the interrogation of the affordance model of the human to determine which agent (either human or robot) shall perform the actions at the next level of decomposition, and whether the robot support shall be planned in the execution of the method.
- According to the agent that has been selected by the affordance model of the human, the HTN planner decomposes the task to be executed either by the human or through the robot collaboration.
- When the support of the robot is needed, its action is determined by interrogating the affordance model of the robot. The action is selected according to the environmental conditions (i.e. object's properties) and

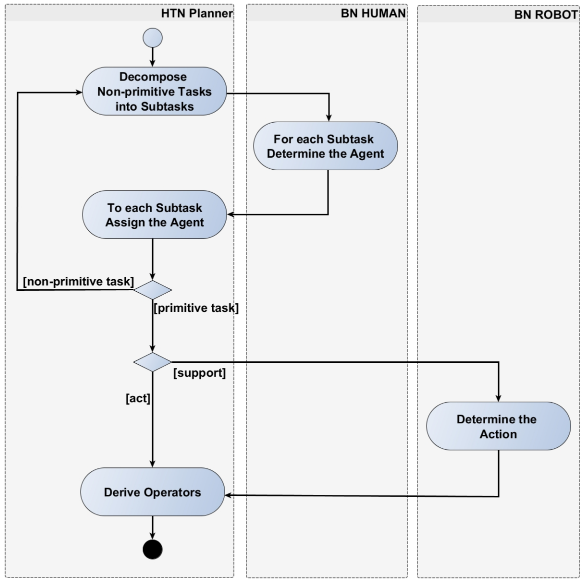


Fig. 4: Information flow between the HTN Planner and the Robot knowledge model (BNs).

the preferences of the human partner (i.e. different ways of obtaining the same effect).

- The actions that shall be performed by the robot are given as inputs to the planner, which identifies the corresponding operators. Actions assigned to the robot are refined using the affordance model of the robot. The pre-condition is modeled in the BN by the object's properties, the post-condition is captured by the desired effect, and the affordance model determines the action to be actually executed by the human.
- When the robot cannot clearly determine who is the actor of the action or the action to be performed (e.g. none of the actions have a probability value that is higher than a threshold), it could use language to resolve the ambiguity.

IV. APPLICATIONS AND EVALUATION

We present the evaluation on the participation of the robot in a shared task with a human co-worker. To showcase the functioning of the proposed planner we have implemented applications for an assembly domain on a Baxter Research robot. We have tested the robotic platform with respect to the possibility of intervening by taking action into a sequence of actions performed by the human. The robot can employ the affordance model to reason on its action possibilities as well as those of the human and offer to help depending on this evaluation.

A. Experimental Setup

To implement the assembly task with the Baxter robot we have developed the necessary software modules for (i) object perception and (ii) action primitives. To fulfill our requirements, we implemented two low-level interfaces (one for each of the robot's arms) able to operate in parallel. A library of high-level actions is surfaced to the planner: on top of executing the action requested, the system is able to provide granular feedback, e.g. by returning a failure if the robot performed an unsuccessful action that was observable by the robot. Inverse kinematics is provided by *TRAC_IK* [17]. The perception system is based on Aruco [18], a fiducial markers library. Each part to be assembled is provided with an unique identifier, that allows for 3D tracking of the part in the robot's operational space thanks to the integration of the visual feedback received by the end effectors' cameras and the robot kinematics.

The system allows for multiple communication layers to interact with the human partner. In this work, we use a subset of the available channels, and specifically: i) a *Feedback* channel is shown in the Baxter's head display, and allows the robot to communicate about its state and intents (see Figure 1); ii) a *Force Interaction* layer detects specific force patterns applied by the human; it is used to achieve natural interactions, e.g. deciding when to release an object during a `pass_object` action; iii) an *Error* channel, triggered by pressing one of the buttons on each of the robot's end effectors, allows the human partner to send error messages if the robot is taking the wrong action or is generally in error.

B. Task Definition

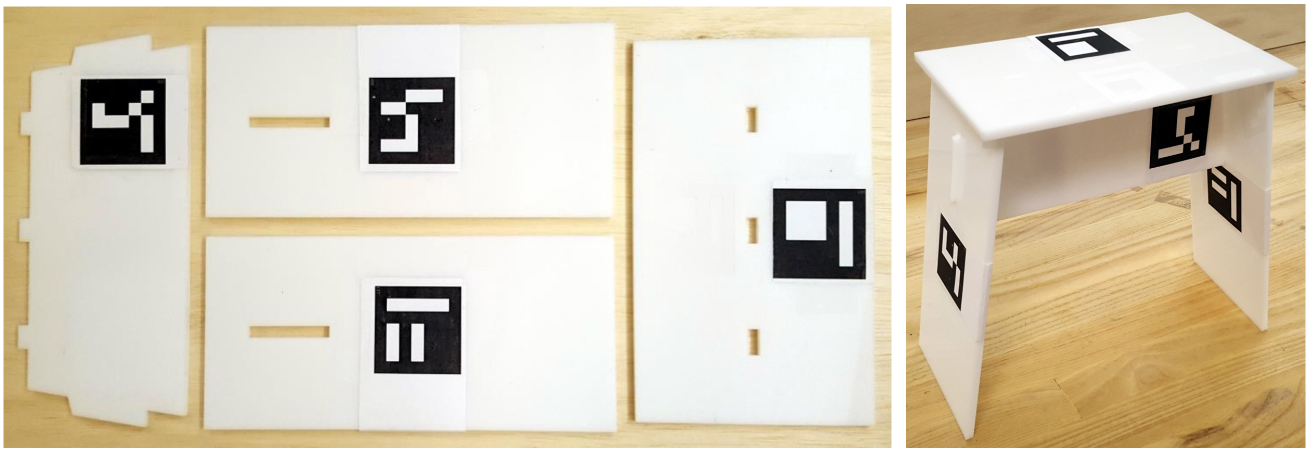
The robot assists a human partner while they are engaged into a collaborative task (cf. Fig. 1). The human is at the workbench while the robot stands on the other side of the table in front of the human. The human and the robot are engaged in the assembly of a stool (Fig. 5), which consists of four parts: Central Frame (`CF`), Left Leg (`LL`), Right Leg (`RL`), Top (`TO`). The constituent parts are distinguished by unique fiducial markers. Each part can be located either on the side of the workbench, or on a table to the left of the robot. The constituent parts are assembled in the workbench area.

C. Bayesian Networks Models of the Human and the Robot

This section describes the structure of the BN models adopted for our application; both models are low-dimensional spaces due to the low complexity of the considered task. The scalability of BNs for modeling complex affordance scenarios has already been shown in other work [19], [20].

The BN model of the human is composed by 4 nodes: *Object Id* represents the unique object's identifier (i.e. `CF`, `LL`, `RL`, `TO`). *Object Position* is the position of the object and it can assume two values (e.g. either the `Workspace` or the `Table` on the robot's side). *Effect* models the desired outcome of the action performed by the human; this node can assume two values (i.e. either `Got` or `Mounted`). *Action* represents the intended behavior of the human that, as explained in Sec III-B.1, can assume two values (i.e. Act or Support).

The BN model of the robot consists of 3 nodes: *Object Id* represents the unique object's identifier (i.e. `CF`, `LL`, `RL`, `TO`). *Effect* is the desired outcome for the action performed by the robot, and it can assume four values:



(a) Constituent parts (b) Assembled stool

Fig. 5: (a) Constituent parts of the stool to be assembled by the human with the support of the Baxter robot: Central Frame (`CF`), Left Leg (`LL`), Right Leg (`RL`), Top (`TO`). (b) Assembled stool.

- `Got` means that the robot has grabbed a part.
- `Passed` means that the robot has given the part to the human partner.
- `Handed over` means that the robot has passed the part from one arm to the other.
- `Held` means that the robot is holding a part to facilitate the assembly task to the human.

Action represents the behavior that the robot has to perform for providing support to the human; this node can have different values that define different ways of performing an action (e.g. `Hold Side`, `Hold Center`). According to the environmental conditions (e.g. object's properties) and the preferences of the human co-worker, the model selects the preferred execution of an action (e.g. actions as `Grab Left`, `Grab Right`, `Grab Center`, `Pass Left`, `Pass Right`, `Hold Side`, `Hold Center`).

Both models have been trained using synthetic data representative of user preferences. On-line training of BNs has been demonstrated in other works [19], [20] and it is hence considered not relevant for the contribution of this work.

D. Task Structure

After the human selects the collaborative task to perform (e.g. assemble a stool), the robot determines the plan that includes the sequence of actions to be performed both by the robot and the human. In our experiment, the HTN planner is pre-configured with the required methods and operators for the execution of the task. According to the environment conditions, the robot can decompose the task of assembling the stool into a sequence of sub-tasks: `Orient Central Frame`, `Attach Legs` and `Attach Top` (cf. Fig. 6). The decomposition of the non-primitive task into sub-tasks requires the application of methods. The parameters associated to the HTN methods, including the current state of the world, specify how to perform the sub-tasks. Such parameters are mapped into an instance of the BN model of the human that in turn selects the action of the human and hence the actor of the sub-task. If the action selected by the BN model is "`Support`" (i.e. a supportive action that involves the robot helping the human partner), then the actor of the sub-task are both the robot and the human. Then, the sub-task is divided into simpler atomic actions and each of them is assigned either to the robot or the human. The primitive tasks `Hand over`, `Hold`, `Get` and `Pass` (cf. Fig. 6) are the behaviors that the robot can perform to support the human partner. The BN model of the robot is interrogated for determining the action to be executed by the robot. The selected actions are given in input to the HTN planner that maps affordances into operators. Indeed, for each primitive task the robot has an internal representation based on affordances that includes information on the user preferences in performing specific actions.

E. Evaluation

We test the capability of the robot to make plans under different conditions. The planner leverages the knowledge of the environment to determine the plan matching the user expectations, both in terms of required support and action execution. Moreover, we test the capability to recover in the event of action failure. The results are presented in the video supplement at `https://youtu.be/d1A-kxW1Rsw`. The performed tests are described below:

- **Test 1: Environment adaptation** This test exercises the BN models, leveraging the user preferences to determine the appropriate behavior (plan) in relation to different environment conditions; specifically, the objects are in different initial positions, and the robot determines when to provide support to the user, and how to manipulate the objects.

  a) CASE I: All the constituent parts of the stool are out-of-reach for the human co-worker. The robot collaborates with the human partner by using the right arm to pass him all the constituent parts, while holding the central frame with the left arm (Fig. 7). The robot

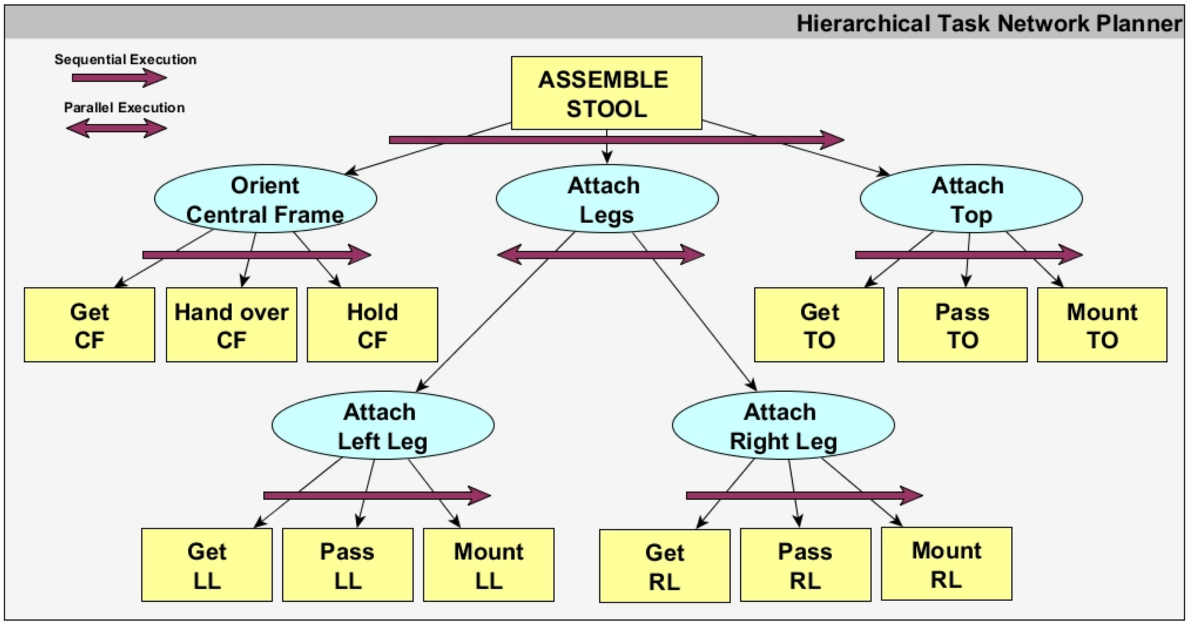


Fig. 6: Task decomposition for the scenario in which all the sub-tasks are performed by the human with the collaboration of the robot.

executes the plan without errors: it selects the proper part to assemble (Fig. 7(a)), it grabs it with the left arm (Fig. 7(b)), it hands it over to the right arm to hold it up to facilitate to the human the mounting of the left (Fig. 7(c)) and right legs (Fig. 7(d)), and it passes the top to complete the stool assembly (Fig. 7(e)).

b) CASE II: Two constituent parts are out-of-reach for the human partner. The robot supports the human co-worker by passing the central frame and the top, and holding the central frame while the human partner mounts the legs. The robot does not intervene while the human gets the constituent parts that are reachable to him. The robot executes the plan without errors.

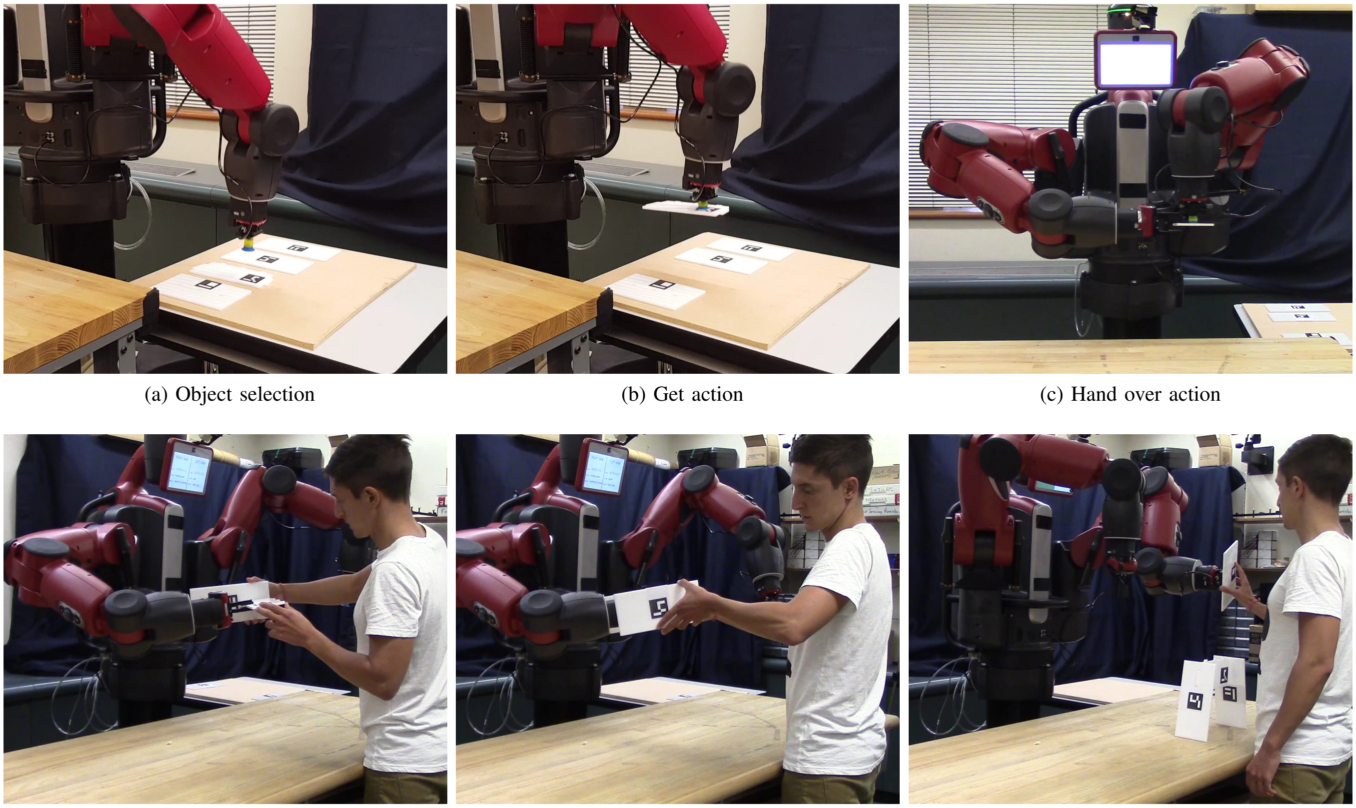
- **Test 2: Action Failure** This test exercises the capability of the planner to recover from the failure of an individual operator; the failed operator is reintroduced in the plan to complete the execution of the task.

  a) CASE I: The robot is presented with an unobservable error, that is it picks a "broken" `CF` (in this experiment, a "broken" part is a part with the correct fiducial marker but with some errors in its shape or affordance). The human communicates the mistake to the robot through the error layer presented in Section IV-A, specifically by pressing the corresponding button on the left arm. The planner re-plans this action. After completing the action of taking the left leg with the right arm (action that is already occurring when the human presses the button) the robot takes again the central frame. This can happen at any point in the sequence and the planner is capable of recovering.

  b) CASE II: The robot is presented with another unobservable failure—in this case fails to take the right leg (we purposely open the vacuum gripper placed on the Baxter's left arm in order to simulate such a scenario). Again, the human partner presses the error button (on the right arm) to notify the failure of the action. The planner performs the same action a second time. If the action is not completed with success after three attempts, the plan fails.

V. CONCLUSION

We presented an affordance-based action planner for the on-line and concurrent collaboration of a human with a robot on a shared task. We applied the proposed affordance-based planner to an assembly task for demonstrating the collaboration of a human worker with the Baxter robotics platform. The proposed planner enabled the robot to: (i) derive a high-level manipulation strategy of a task that requires the performance of a sequence of actions of both the robot and the human and (ii) decide when to intervene by taking action into the sequence of actions performed by the human. For building plans shared between the robot and the human we exploited the knowledge represented by affordance models. The proposed planner demonstrated two important features: (i) the planning of concurrent actions and (ii) the reaction to action failure (i.e. on-line planner). For the planning of concurrent actions, we assumed that each robot arm is an independent agent that can perform different actions; this increases the level of support that the robot can provide, and improves the working conditions of the human. The on-line planner allows to dynamically adapt the plan during the execution, to account for variations in the environment or in general for failures in the execution of a task by the robot. In this work, the planner will reiterate the failed operation; more complex re-planning strategies will be integrated in future evolution of the proposed framework.



(a) Object selection (b) Get action (c) Hand over action

(d) Hold action while human mounts LL (e) Hold action while human mounts RL (f) Pass action for attaching TO

Fig. 7: Temporal snapshots of the robot and the human co-worker during Test1 (CASE I), see Sec. IV-E.

REFERENCES

[1] B. Hayes and B. Scassellati, “Effective robot teammate behaviors for supporting sequential manipulation tasks,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 6374–6380.

[2] C. Breazeal, A. Brooks, D. Chilongo, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, and A. Lockerd, “Working collaboratively with humanoid robots,” in *Humanoid Robots, 2004 4th IEEE/RAS International Conference on*, vol. 1, Nov 2004, pp. 253–272 Vol. 1.

[3] J. J. Gibson, *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979, includes indexes.

[4] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[5] K. Williams and C. Breazeal, “A reasoning architecture for human-robot joint tasks using physics-,social-, and capability-based logic,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 664–671.

[6] 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.

[7] T. Lozano-Perez, J. Jones, E. Mazer, P. O’Donnell, W. Grimson, P. Tournassoud, and A. Lanusse, “Handey: A robot system that recognizes, plans, and manipulates,” in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 843–849.

[8] M. Toussaint, “Logic-geometric programming: An optimization-based approach to combined task and motion planning,” in *International Joint Conference on Artificial Intelligence*, 2015.

[9] D. S. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman, “Shop2: An htn planning system,” *J. Artif. Intell. Res.(JAIR)*, vol. 20, pp. 379–404, 2003.

[10] I. R. Nourbakhsh, “Using abstraction to interleave planning and execution,” in *Proceeding of the third biannual world automation congress*, 1998.

[11] L. P. Kaelbling and T. Lozano-Pérez, “Hierarchical task and motion planning in the now,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1470–1477.

[12] B. Hayes and B. Scassellati, “Autonomously constructing hierarchical task networks for planning and human-robot collaboration,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5469–5476.

[13] I. Awaad, G. K. Kraetzschmar, and J. Hertzberg, “The role of functional affordances in socializing robots,” *International Journal of Social Robotics*, vol. 7, no. 4, pp. 421–438, 2015.

[14] O. Obst, “Using a planner for coordination of multiagent team behavior,” in *International Workshop on Programming Multi-Agent Systems*. Springer, 2005, pp. 90–100.

[15] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, “Ikeabot: An autonomous multi-robot coordinated furniture assembly system,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 855–862.

[16] B. Hayes and B. Scassellati, “Discovering task constraints through observation and active learning,” in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 4442–4449.

[17] 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.

[18] 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.

[19] M. Lopes, F. S. Melo, and L. Montesano, “Affordance-based imitation learning in robots,” in *Intelligent Robots and Systems. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 1015–1021.

[20] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, “Learning object affordances: From sensory–motor coordination to imitation,” *Robotics, IEEE Transactions on*, vol. 24, no. 1, pp. 15–26, 2008.