MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

E-UHTP: An Extended Human-Aware Task Planner for Communication-Free Collaborative Assembly

Giacomuzzo, Giulio; Pegoraro, Giulia; Terreran, Matteo; Ghidoni, Stefano; Carli, Ruggero; Romeres, Diego

TR2025-066 May 28, 2025

Abstract

We propose E-UHTP, an enhanced task planner for collaborative assembly in communication-free human-robot environments. E-UHTP extends User-aware Hierarchical Task Planning (UHTP) by supporting joint actions, failure recovery, and online replanning. We introduce an automatic Hierarchical Task Network (HTN) generation method from annotated video demonstrations. Experimental simulations in multiple assembly scenarios demonstrate the improved performance, flexibility, and robustness of E-UHTP over baseline planners.

IEEE International Conference on Robotics and Automation (ICRA) - Workshop on The Future of Intelligent Manufacturing: From innovation to implementation 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

E-UHTP: An Extended Human-Aware Task Planner for Communication-Free Collaborative Assembly

Giulio Giacomuzzo[1], Giulia Pegoraro[1], Matteo Terreran[1], Stefano Ghidoni[1], Ruggero Carli[1], Diego Romeres[2]

Abstract— **We propose E-UHTP, an enhanced task planner for collaborative assembly in communication-free human-robot environments. E-UHTP extends User-aware Hierarchical Task Planning (UHTP) by supporting joint actions, failure recovery, and online replanning. We introduce an automatic Hierarchical Task Network (HTN) generation method from annotated video demonstrations. Experimental simulations in multiple assembly scenarios demonstrate the improved performance, flexibility, and robustness of E-UHTP over baseline planners.**

I. Introduction

Technological innovation has profoundly reshaped the industrial landscape, integrating advanced automation and artificial intelligence into manufacturing processes. Among the most promising developments is Human-Robot Collaboration (HRC), a paradigm that combines the distinctive abilities of human workers, such as dexterity and decision-making, with the precision, strength, and repeatability of robots. Unlike traditional industrial robots, collaborative robots, or cobots, are designed to safely share workspaces with humans, enhancing workflow efficiency without the need for physical barriers.

A particularly significant area for HRC is collaborative assembly. Assembly processes typically involve long sequences of subtasks, alternating between heavy manipulations and delicate operations. These processes demand efficient coordination between humans and robots, continuous monitoring of actions, and dynamic task planning to manage the sequential dependencies and constraints. However, most existing frameworks rely on strict predefined plans [1]–[3] or explicit communication between partners [4]–[6], which can reduce naturalness, flexibility, and overall collaboration efficiency. In real human teams, intentions are often inferred by observing actions rather than through constant verbal exchanges, and ideally, robots should exhibit similar adaptive capabilities.

An important step in this direction is the User-aware Hierarchical Task Planning (UHTP) framework [7], a communication free solution for human-robot collaborative assembly. UHTP enables the robot to observe the human partner and autonomously plan its actions without relying on predefined paths or explicit human models. By continuously monitoring the state of the assembly and the human's actions, UHTP selects optimal actions that respect task constraints while leaving the human free to act naturally. This improves adaptability and interpretability, addressing the limitations of methods that force the human to follow prescribed plans or require constant communication.

While UHTP represents a valuable advance, it does not address two critical aspects of collaborative assembly: the execution of joint actions and the management of failures. Joint actions, where the human and robot perform simultaneous or interdependent tasks, frequently occur in realistic assembly environments but are unsupported in the original UHTP formulation. Additionally, UHTP assumes a flawless execution of tasks, whereas in practice, unexpected events and failures are inevitable and need to be handled effectively to maintain workflow efficiency.

Moreover, UHTP relies on a Hierarchical Task Network (HTN) [8] to represent and manage collaborative assembly processes. HTNs organize actions to be executed in a multi-level structure that encodes sequential or independent relationships between subtasks. Due to their modularity, clarity, and interpretability, they are particularly effective for planning, in particular when generating and updating task plans based on ongoing observations of the human partner's actions. Nonetheless, manually deriving HTNs remains a laborious, expertise-dependent process, especially for complex assemblies involving numerous actions and complex dependencies.

This work proposes an extension of UHTP that overcomes the aforementioned limitations. The first contribution is the development of a novel algorithm for automatically deriving HTNs directly from annotated video demonstrations, streamlining the creation of structured, hierarchical task descriptions for complex assemblies. Building upon this, the second and core contribution involves extending the UHTP planner to handle joint actions, enabling the robot to perform synchronized or complementary operations with the human partner. Furthermore, a failure management mechanism is integrated into the planner, allowing the robot to autonomously detect, manage, and replan around unexpected events without requiring explicit communication.

The proposed system retains the optimization-based, interpretable nature of UHTP while enhancing its flexibility and robustness in dynamic collaborative settings. Simulated assembly experiments confirm that the extended framework improves task success rates, adaptability, and efficiency compared to baseline approaches. Together with random toy cases, we simulate the assembly of an IKEA chair as one of the use cases for the proposed algorithm, whose HTM is reported in Fig. 2.

[1]Department of Information Engineering, Università di Padova, Italy
[2]Mitsubishi Electric Research Laboratories, Cambridge, MA, USA
Corresonding contact: romeres@merl.com

II. Background

Assembly problems can be seen as a hierarchical set of actions, which must be executed in compliance with a set of ordering constraints. We assume actions to be either individual or joint actions. Individual actions can be executed by a single agent, either the human or the robot. Joint actions must be executed by both agents together. Moreover, we assume the human to be an uncontrolled agent, to which the robot should adapt.

A. Hierarchical Task Networks

Hierarchical Task Networks are a well-established formalism for representing procedural knowledge, particularly suited for planning problems. An HTN decomposes a complex assembly task into a hierarchy of simpler subtasks. The network typically consists of non-primitive tasks, which need further decomposition, and primitive tasks, which correspond to executable actions. Subtasks can be ordered, imposing sequential constraints, or unordered, allowing for parallel execution or flexible ordering.

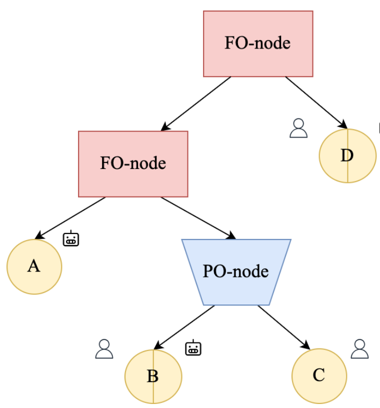

Fig. 1. Example HTN. Nodes represent tasks (internal nodes) or primitive actions (leaves). Ordering constraints are expressed through Partially Oriented (PO) or Fully Oriented (FO) nodes.

We represent HTNs as trees. A pictorial example of HTN is reported in Figure 1. The root represents the overall assembly goal. Internal nodes can be non-primitive tasks requiring decomposition or control nodes specifying execution constraints. We utilize two primary control nodes: Fully Ordered (FO) nodes, whose children must be executed sequentially from left to right, and Partially Ordered (PO) nodes, whose children can be executed in any order or potentially in parallel, provided their individual preconditions are met. Primitive actions form the leaves of the tree. An assembly is considered completed when all the primitive actions have been completed.

B. User-aware Hierarchical Task Planning (UHTP)

The UHTP framework [7] adapts HTN planning for communication-free collaborative assembly. It extends the standard HTN structure by assigning potential actors (human or robot) and associated costs (e.g., execution time) to primitive actions. If an action can be performed by either agent, it is represented by a Decision (D) node (see Figure 2) with agent-specific primitive action nodes as children.

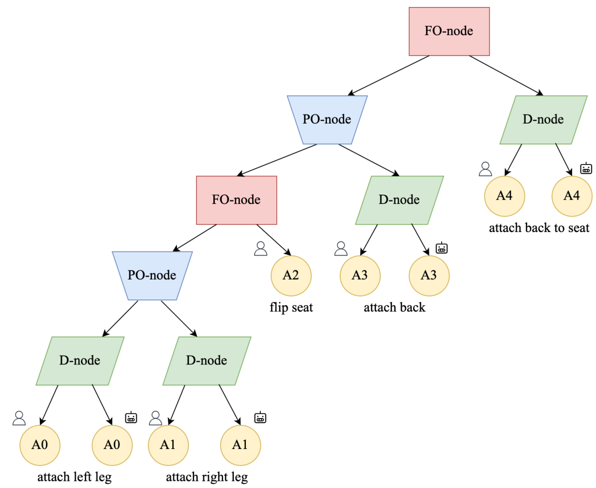

Fig. 2. Example HTN representing a chair assembly. Nodes represent tasks (internal nodes) or primitive actions (leaves). Ordering constraints are expressed through Partially Oriented (PO) or Fully Oriented (FO) nodes.

The core idea of UHTP is to enable the robot to plan its actions by considering the current human's actions and all their possible future choices, without needing to predict them accurately or rely on communication. The workflow of UHTP is reported in Algorithm 1. The assembly HTN ($\mathscr{H}$) is first extended by adding D-nodes and computing costs. Costs are calculated recursively bottom-up. Primitive node cost is its intrinsic cost (e.g. expected time duration). Decision node cost is a probability-weighted sum of its children's costs, assuming the agents' probabilities as known parameters. FO node cost is the sum of its children's costs. PO node cost calculation requires considering bounds; the cost is typically between the maximum child cost (perfect parallelism) and the sum of child costs (sequential execution), often approximated or calculated based on expected overlap.

When the human completes an action, the HTN is updated by pruning branches corresponding to the completed action and any now-invalid alternatives. When the robot becomes idle, its next action is selected by evaluating all the currently available robot actions, pruning the tree for each potential choice, calculating the aggregated cost of the remaining subtree, and finally selecting the action leading to the minimum expected total cost.

Then, the robot keeps observing the human's actions and uses this information to prune the HTN, effectively updating the current state of the assembly. Then, it calculates the expected cost-to-go for completing the remaining tasks for each possible sequence of actions. Finally, it selects the action that minimizes this expected cost, thereby adapting its strategy based on the human's progress and choices. This allows for mutual adaptation, where both agents influence the task execution flow without explicit coordination signals.

The innovation of the UHTP framework is represented by the combination of the Action Optimization and the HTN tree pruning. Optimization over trees, indeed, could become particularly expensive, in particular for long-lasting and complex assemblies. Removing all the impossible sequences each time an action is selected represents a simple yet effective solution to maintain the complexity tractable. Due to space constraints, we refer the interested reader to [7] for a detailed description of the UHTP methods.

Algorithm 1 UHTP program flow

function UHTP($\mathscr{H}$)
  $\mathscr{H}^{UHTP} \leftarrow$ ASSIGNACTIONS($\mathscr{H}$)
  COMPUTECOSTS($\mathscr{H}^{UHTP}$)
  $a_h \leftarrow idle$
  $a_r \leftarrow idle$
  while $\mathscr{H}^{UHTP}$ is not empty **do**
    $a'_h \leftarrow$ ACTIVITYRECOGNITION()
    if $a'_h \neq a_h$ **then**
      $\mathscr{H}^{UHTP}$.REMOVE(a_h)
      PRUNEBRANCHES($\mathscr{H}^{UHTP}$), a'_h)
      $a_h \leftarrow a'_h$
    end if
    if robot is idle **then**
      $\mathscr{H}^{UHTP}$.REMOVE(a_r)
      $a_r \leftarrow$ SELECTMINACTION($\mathscr{H}^{UHTP}$)
      PRUNEBRANCHES($\mathscr{H}^{UHTP}$)
      EXECUTE(a_r)
    end if
  end while
end function

III. METHODOLOGY

In this Section, we present our contributions: an algorithm for HTN construction from demonstrations and the E-UHTP.

A. HTN Construction from Annotated Videos

Automating HTN generation reduces manual effort and can capture realistic task structures. Our approach takes as input a dataset of annotated videos representing successful assembly executions. Each video is represented as a sequence of tuples, where each tuple contains the actions being performed simultaneously by the human and the robot, or 'idle' if an agent is inactive. Each action is described by its name, its duration in time units, and the agent performing it. Joint actions are represented by identical action names for both agents in a tuple. An example of a tuple regarding the chair assembly example is reported below.

```
(["attach left leg",[3],["human"]],
["attach right leg",[2],["robot"]])
```

The algorithm proceeds in two main stages. First, it extracts action requirements, that is, for each primitive action observed in the dataset, the algorithm determines the set of other actions that must be completed before it can start. To this aim, for each video sequence, a list of actions completed so far is maintained. When an action occurs, the set of already completed actions represents the potential set of preconditions. By analyzing multiple videos showing different valid execution paths, the algorithm identifies the necessary preconditions for each action as the intersection of the sets of completed actions observed just before the action starts. An example of the results returned by this process is reported below for one of the actions of the chair example. The list reports the action name, the nominal durations extracted from the videos, the agents able to perform the action, and the requirements.

```
["attach back to seat",
[5,7],
["human","robot"],
["attach right leg",
  "attach left leg",
  "attach back",
  "flip seat"]]
```

Second, the algorithm constructs the HTN tree structure based on the requirements. Note that, while the requirements set derived from an HTN is unique, multiple valid HTN structures might be possible from a given set of requirements, depending on how subtasks are grouped. Our approach constructs one of the possible HTNs. The construction proceeds bottom-up from the leaves (primitive actions). Actions with no requirements can be placed first. Actions whose requirements are fully met by the actions already placed in a partially constructed subtree can then be added. If an action has been seen to be performed by both a human and a robot, a decision node is created. When multiple actions become available simultaneously (i.e., their requirements are met by the same set of completed actions), they are grouped under a PO node if their requirements do not impose a sequence between them, signifying potential parallelism. If the requirements dictate a specific order, instead, they are placed sequentially under a FO node. This process recursively builds the tree until the root node, representing the final assembly goal, is reached.

B. Extended UHTP Framework Implementation

Our task planning implementation builds upon the UHTP concepts, extending them with failure recovery and joint action handling.

Failure Recovery is implemented to handle unexpected action failures. We assume failures to be detected by the monitoring system. Furthermore, for simplicity, we assume that each primitive action has a corresponding recovery action to be executed in case of failure. The problem of making the robot autonomous in deciding how to handle failures is an attractive yet very complex aspect, which is left as possible future work. To handle failures, we introduce a novel node type called Recovery (R) node. Upon failure detection, the failed action node is temporarily removed from the HTN, and an R-node is inserted into the HTN as a child of the failed action's parent. This R-node has its own associated cost and duration and may represent tasks like picking up a dropped part or correcting a misalignment. The completion of the recovery task triggers the re-insertion of the original failed action node into the HTN, allowing it to be attempted again when its requirements are met. The framework ensures that subsequent dependent actions cannot proceed until the recovery and the original action are successfully completed. The mechanism described above makes the HTN vary dynamically, according to the outcomes of the monitoring system, which introduces more flexibility and robustness to the framework. Note that this mechanism can be easily extended to handle any kind of unpredictable event, such as human changes of mind or agents' inabilities to complete a task.

Joint Actions are incorporated to account for tasks requiring simultaneous effort. In order to address joint actions, in the E-UHTP framework, we enrich primitive actions with an additional agent, hereafter denoted joint. We assume that only the human agent can initiate a joint action. When the human initiates a joint action, the system checks the robot's status. If the robot is idle, it immediately starts executing the same joint action, synchronized with the human. If the robot is busy with another task, it completes the ongoing task first and then joins the human in to perform the joint action. Failures during joint actions are handled by the standard recovery mechanism, potentially requiring both agents to perform the recovery together.

IV. Experimental Evaluation

We conducted simulation experiments to validate the proposed methods. To simulate the assembly tasks, we modeled the human as a random agent. The duration of the actions is modeled as a Gaussian distribution with nominal values of the action duration as mean and a standard deviation set to 5 % of the nominal duration.

A. Failure Recovery Robustness Evaluation

To assess the robustness introduced by the failure recovery mechanism, we simulated task executions under varying conditions of uncertainty. We considered the chair assembly task, systematically increasing the probability of failure for each primitive action from 10% up to 50%. For each probability level, 100 runs were performed, recording the total task completion time. Table I reports the completion times. As expected, the mean and variance of the completion time increase with higher failure probabilities, reflecting the time spent on recovery actions and re-attempts. Despite the increased execution time, the extended UHTP framework successfully completed the assembly task in all simulation runs across all tested failure rates and task complexities. This demonstrates that the implemented recovery process ensures task completion, enhancing the framework's robustness.

TABLE I
E-UHTP Completion Time at Different Failure Probabilities on the Chair Example

Failure Probability [%]	10%	20%	30%	40%	50%
Task Completion Time [s]	27	28	34	40	46

B. Planning Policy Comparison

We evaluated the efficiency of the E-UHTP framework against two baseline policies: a Greedy policy, where the robot selects the available action with the lowest immediate cost (duration), and a Random policy, where the robot randomly selects among its currently available actions.

We ran 1000 simulations for each policy on three different tasks: the chair assembly and two randomly generated tasks with 16 and 32 actions, respectively, featuring varying requirement complexities. Within the randomly generated tasks, we also randomly included joint actions. Table II presents the mean task completion times. Across all tested scenarios, the E-UHTP policy consistently achieved the lowest average completion time. While the Greedy policy performed better than Random, it was often suboptimal, as minimizing immediate cost does not guarantee minimizing total assembly time. The E-UHTP policy's ability to consider the aggregated cost of the remaining task tree allowed for making more globally efficient decisions, effectively coordinating with the unpredictable human agent to minimize overall duration. The standard deviation was also generally comparable or lower for E-UHTP, suggesting consistent robustness to randomization.

TABLE II
Task Completion Time Comparison

Policy	**16 Actions**	**32 Actions**	**Chair**
Greedy Policy	178.75 ± 11.10	340.71 ± 14.21	22.52 ± 0.65
Random Policy	187.38 ± 12.33	344.26 ± 14.60	22.95 ± 0.42
E-UHTP Policy	$\mathbf{172.26 \pm 7.08}$	$\mathbf{331.10 \pm 11.32}$	$\mathbf{22.16 \pm 0.26}$

V. Conclusions

This paper addressed key challenges in task representation and planning for human-robot collaborative assembly within a communication-free setting. We introduced two main contributions to enhance the flexibility and robustness of such systems. First, we developed and validated an algorithm capable of automatically constructing Hierarchical Task Network representations from annotated video demonstrations. Second, we presented an extension of the state-of-the-art UHTP framework, named E-UHTP. E-UHTP incorporates mechanisms for handling joint actions and for recovering from action failures. The framework maintains the core UHTP principle of optimizing robot actions based on aggregated costs while treating the human as an unpredictable agent, thus avoiding restrictive human modeling or explicit communication requirements.

References

[1] F. Fusaro, E. Lamon, E. De Momi, and A. Ajoudani, "A human-aware method to plan complex cooperative and autonomous tasks using behavior trees," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 522–529.

[2] K. Darvish, F. Wanderlingh, B. Bruno, E. Simetti, F. Mastrogiovanni, and G. Casalino, "Flexible human–robot cooperation models for assisted shop-floor tasks," in *Mechatronics*, vol. 51. Elsevier, 2018, pp. 97–114.

[3] 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, 2016.

[4] A. Roncone, O. Mangin, and B. Scassellati, "Transparent role assignment and task allocation in human robot collaboration," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1014–1021.

[5] 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 (HRI '11)*. ACM, 2011, pp. 29–36.

[6] G. Milliez, R. Lallement, M. Fiore, and R. Alami, "Using human knowledge awareness to adapt collaborative plan generation, explanation and monitoring," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 43–50.

[7] K. Ramachandruni, C. Kent, and S. Chernova, "UHTP: A user-aware hierarchical task planning framework for communication-free, mutually-adaptive human-robot collaboration," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 4, pp. Article 43, 1–28, 2023.

[8] K. Erol, J. A. Hendler, and D. S. Nau, "UMCP: A sound and complete procedure for hierarchical task-network planning," in *Proceedings of the 2nd International Conference on AI Planning Systems (AIPS'94)*, 1994, pp. 249–254.