Collaborative Human-Robot Hierarchical Task Execution with an Activation Spreading Architecture

Bashira Akter Anima[1], Janelle Blankenburg[1], Mariya Zagainova[1], S. Pourya Hoseini A.[1], Muhammed Tawfiq Chowdhury[1], David Feil-Seifer[1], Monica Nicolescu[1], and Mircea Nicolescu[1]

Department of Computer Science & Engineering, University Nevada, Reno, Reno, NV 89557, USA banima@nevada.unr.edu

Abstract. This paper addresses the problem of human-robot collaborative task execution for hierarchical task plans. The main contributions are the ability for dynamic allocation of tasks in human-robot teams and opportunistic task execution given different environmental conditions. The human-robot collaborative task is represented in a tree structure which consists of sequential, non-ordering, and alternative paths of execution. The general approach to enable human-robot collaborative task execution is to have the robot maintain an updated, simulated version of the human's task representation, which is similar to the robot's own controller for the same task. Continuous peer node message passing between the agents' task representations enables both to coordinate their task execution, so that they perform the task given its required execution constraints and they do not both work on the same task component. A tea-table task scenario was designed for validation with overlapping and non-overlapping sub-tasks between a human and a Baxter robot.

1 Introduction

The fast pace of advancements in the development of autonomous robotic systems opens new possibilities for the use of robots in daily tasks, holding a significant potential for improving the quality of our lives. While autonomy and the ability of robots to perform complex tasks have significantly improved, the challenges of operating in collaborative domains prevent current robotic systems from working effectively alongside with people as collaborators and assistants. The focus of the proposed work is to develop a control architecture that enables robots and humans to work collaboratively on a joint task that has a complex hierarchical structure and multiple types of execution constraints.

The underlying assumption is that both the robot and the human have knowledge of the requirements of the task. However, there is no pre-defined allocation that indicates what the human or the robot should do, and both teammates are allowed to work on any aspect of the task, as long as they obey the execution constraints imposed for the task (e.g., ordering of steps). As a result, the robot's

decision making process (i.e., deciding what part of the task to work on) is tightly interconnected with its ability to understand the human teammate's goals and intentions. For this, each robot needs to take into account what are the overall (sub-)goals of the task, and also which (sub-)goals are already being worked on by the human. In a team comprising only of robots, such information may be transmitted through direct communication; when interacting with human users, a robot would need to rely on direct observations (e.g., using cameras) in order to track the humans' actions.

In this paper, we propose a solution where the robot uses its own task representation (e.g., controller) both to plan its own future actions, and to keep track of its human teammate's current and future goals. The general solution is as follows: the robot maintains a duplicate representation of the task controller for the human teammate, representing the human's mental model of the task. This second representation "runs" in parallel with the robot's own representation, and the status of various nodes in the human's task (e.g., *working*, or *done*) is updated by the robot using its camera. Peer nodes on both the robot's and the human's controllers continuously exchange messages that communicate their status information, enabling the robot to infer what part of the task the human is working on. The robot decides its next action based both on the constraints of the defined task and the behavior of the human partner.

2 Related Work

Collaboration between robots and humans is crucial to the effective utilization of modern robots in the real world. Our experiments focus on the capability of a robot's identification of human intention while working collaboratively with a human. Much prior work has been done in this area. Intent recognition encompasses many domains, including: entertainment [1]; museum documents [2]; personal assistants [3]; health care [4]; space exploration [5]; police SWAT teams [6]; military robotics [6]; and rescue robotics [7]. The proposed work demonstrates the ability for dynamic allocation of tasks in human-robot teams based on intent recognition, while also observing hierarchical constraints.

Approaches exist for recognizing human intent. A recognition task was categorized into two categories: explicit intention communication and implicit intention communication, and using weighted probabilistic state machines were utilized [8]. Recurrent Convolutional Neural Networks (RCNNs) [9] and Neural networks [10] were used to detect human intention, and an online estimation method was developed to deal with the nonlinear and time-varying property of a limb model. Human-aware motion planning was examined in [11] and [12]. The ability of a robot to work with a human in close proximity [13] without colliding with the human was demonstrated. A Gaussian Mixture Model (GMM) representation [14] of a human's motion was used. In our work, collision avoidance after collision detection has been emphasized unlike other works that focused on avoiding collision based on predefined mechanisms.

Given detected intent, it is an open question whether and when a robot should take initiative during joint human-robot task execution [15]. In this work, robot-initiated reactive assistance triggers the robot's help when it senses that the user needs help and robot-initiated proactive assistance makes the robot help whenever it can. In our architecture, we combined the processes of the robot's recognition of human actions and it's decision making to determine when it should take initiative during a human-robot joint task.

A collaborative robot should be able to execute complex tasks, be aware of its teammates' goals and intentions, as well as be able to make decisions for its actions based on this information. Recent work addresses these challenges using a probabilistic approach for predicting human actions and a cost based planner for the robot's response [16]. Tasks are represented as Bayes networks and prediction of human actions is performed using a forward-backward message passing algorithm in the network. This inference process is however dependent on knowledge of the full conditional probability table for the task, which increases computational complexity for large tasks and limits adaptability to changes in the task at run-time. This approach has been extended in [17], with a new task representation that can encode tasks with multiple paths of execution. The initial representation for the task is a compact AND-OR tree structure, but for action prediction and planning, it has to be converted into an equivalent Bayes network, which has to explicitly enumerate all possible alternative paths.

Our task tree representation includes a THEN-AND-OR tree structure which further allows for sequential, alternative paths of execution, and non-ordering constraints. Additionally, our approach is able to choose actions based on a human's intent without having to enumerate all possible alternative paths.

3 Human-Robot Collaborative Architecture

3.1 Hierarchical Task Representation

In this work, we augmented our robot control architecture that enables the system to encode tasks involving various types of constraints such as sequential (THEN), non-ordering (AND), and alternative paths of execution (OR) [18]. Tasks are represented in a tree structure where leaf nodes represent tasks to be completed and behavior nodes represent the hierarchical relationships between those tasks. An example task for arranging a tea table scenario is shown in Fig. 1.

In order to execute a controller represented by such a hierarchical task, each node in the architecture maintains a state consisting of several components: **1) Activation Level:** a number provided by the node's parent and represents the priority placed on executing and finalizing a given node, **2) Activation Potential:** a number representing the node's perceived efficiency, which is sent to the parent of the node, **3) Active:** a boolean variable that is set to true when the node's activation level exceeds a predefined threshold, indicating that the behavior is currently executing, and **4) Done:** a boolean variable that is set to true when the node has completed its required work. The above state information

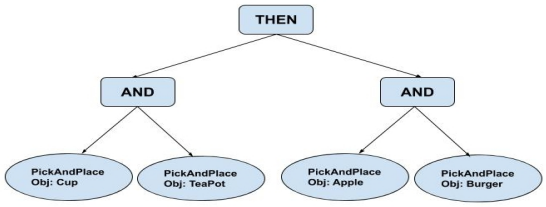


Fig. 1: Hierarchical Task Representation

is continuously maintained for each node and is used to perform top-down and bottom-up activation spreading that ensures the proper execution of the task given the constraints.

To execute a task, *activation spreading messages* are sent from the root node of a task toward its children to spread the *activation level* throughout the task tree. At the same time, each node sends its current state to its parent node as *status messages* to spread the *activation potential* throughout the tree in a bottom-up fashion. An update loop is run at each cycle which helps to maintain the state of each node in the task structure. This loop performs a series of checks of the node's state and updates the various components of the state accordingly.

The controller architecture scales to multiple robots by maintaining a copy of the task tree for each robot noting when that robot is currently working on a behavior, when a robot has completed one, and the activation potential and level for each robot and each behavior. Message passing between peer nodes (equivalent nodes across all robots' copies of the task tree) allow each robot to represent the complete task status, not just its own view. The full details of this approach are presented in [18].

3.2 Human-In-The-Loop Hierarchical Architecture

In order to extend the previously developed architecture described in Section 3.1 from the multi-robot domain to the human-robot domain, several adjustments must be made. The robot can perform a task with a human instead of another robot by maintaining an updated, simulated version of the human's task representation. The person completes the task with the same constraints as the robot. Message passing between peer nodes of the human's and robot's task representation enables the task execution to perform as in the robot-robot scenario.

If the human's sub-task can be inferred, the corresponding node's activation potential in the human's architecture will be increased making the node *active*. As a result, the robot will be able to know what the human is working on. For task execution we distinguish between the following two cases:

1. The human and the robot choose to work on *non-overlapping tasks* in Fig.1. If the human and the robot decide to work on the cup and the teapot respectively, the robot will infer that its sub-task is safe to continue by checking the status of the peer node of the teapot on the human's controller.

2. The human and robot decide to work on *the same sub-task* in Fig. 1. If both agents decide to work on the cup, the node status will indicate to the robot that the human is also working on this sub-task. The robot will initiate a dialogue in order to negotiate the conflict. A *dialogue* topic and *issue* topic to each corresponding node are added to the architecture to initiate the dialogue.

The likelihood that the person is intending to pick up each object based on the updated hand position for each frame is published as an object status message. The behavior node of an object in the human architecture will be updated based on the value of the object status message for each object.

3.3 Human Intention Recognition

During execution of the task, the robot continuously updates the hand position of the human as shown in Fig. 2. By finding the largest skin contour in the image frame, we are able to detect the position of the human hand because the only skin in the robot's view is the hand.

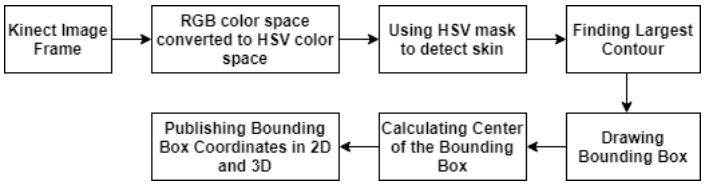


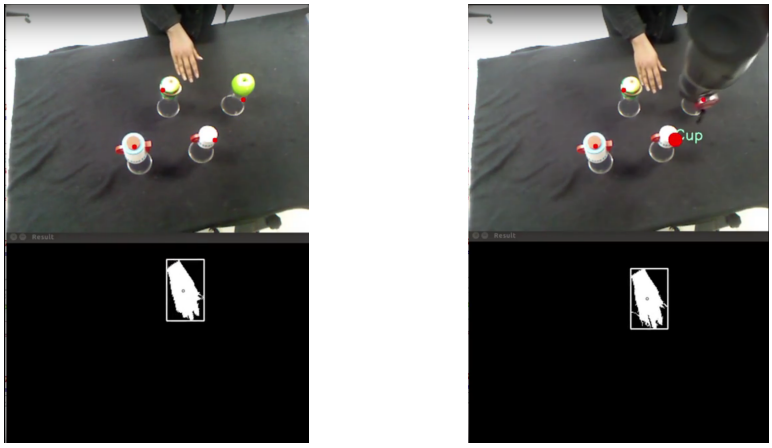
Fig. 2: A step-by-step description of the continuous hand detection system from the Kinect image frame to infer the human intention

From the motion of the hand, we calculate similarity score (*SimScore*), chance score (*Chance*), started value (*Started*) and done value (*Done*) for each object.

– **Similarity Score:** The similarity score (*SimScore*) for each object is calculated for the updated hand position ($h_{x,y,z}$) in the frame. The initial normalized vector between the initial hand position ($h_{X,Y,Z}$) and an object's position ($obj(i)_{x,y,z}$) are calculated for each object $i \in 1, ..., n$. For each new hand position, the cosine similarity between the initial normalized vector and the updated normalized vector are calculated and stored in the *SimScore* list as shown in equation 1.

$$SimScore_i = Cosine\_Similarity(V_{X_i,\hat{Y_i},Z_i}, V_{x_i,\hat{y_i},z_i}) \tag{1}$$

where $V_{X_i,\hat{Y_i},Z_i}$ and $V_{x_i,\hat{y_i},z_i}$ are the initial normalized vector and updated normalized vector for object $i \in 1, ..., n$.



(a) (b)

Fig. 3: Human intention system with the contour of the hand detection (a) The system hasn't detected the intention yet (b) The system is detecting the intention with a red circle on the object.

- **Chance:** The *Chance* value for the object that has the highest *SimScore* is incremented for every new hand position. If multiple objects have the same maximum score, the *Chance* value will be incremented for all of them. In this situation, the *Chance* value of the object which had the highest similarity score in the previous iteration will instead be incremented twice.
- **Started:** A Boolean variable which is initially 0 for each object; it will be set to 1 if it is inferred that the human is going for the object by checking the maximum *Chance* value.
- **Done:** A Boolean variable that will be initially 0 for each object; it will be set to 1 if the task for the object is completed by the human.

The above information (*Chance*, *Started*, and *Done*) is contained in the object status messages which are published to each object's dedicated status topic using ROS [19]. The messages allow the human architecture to activate an object node when the *Started* value is 1.

3.4 Collision Detection and Handling

In most human-robot collaborative tasks, there can be *collisions* where both the human and the robot can go for the same object at the same time. Collisions must be handled for smooth collaboration between human and robot. As mentioned before, each node of each agent's task tree is updated continuously with the status of its corresponding node of the other agent. If both the human and robot are going to the same object simultaneously, then the status of both nodes will be *active*, which will trigger a collision.

Fig. 3 shows the human hand going for the cup during the task. The system hasn't detected the intention yet in Fig. 3a. However, in Fig. 3b the human's intention can now be inferred and is being shown with a red circle on the object.

If a collision is detected, a ROS message will be published to the corresponding node's *issue* topic which will enable the callback function to publish a ROS message to the *dialogue* topic. This initiates the negotiation between the robot and the human. The robot will ask, "It appears that you are going to grab the (Object Name). Should I grab the (Object Name)?" If the human replies "Yes" then the robot will answer "Alright I will place the (Object Name)." The robot will then continue on its path to pick and place the object, while the human will instead go for the next available object in the task tree. If the human replies "No," then the robot will answer "Okay, then please place the (Object Name). Thank you." It will then let the human finish the pick and place task and instead go for the next object according to the task tree.

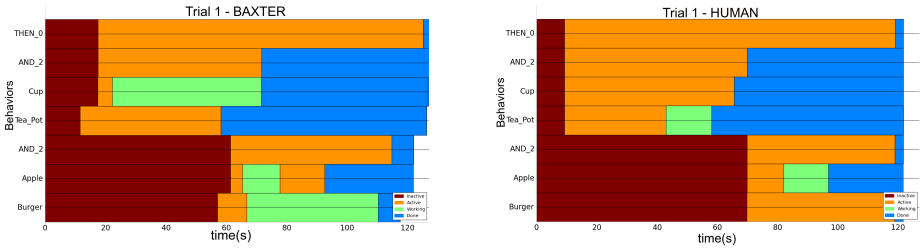
4 Experiment Design

To demonstrate the capabilities of this augmentation of the architecture, a distributive task between a human and a robot was designed. The task was performed in a lab environment with a human and a Baxter humanoid robot standing on opposite sides of a table containing the objects as shown in Fig. 4. The 3D location of each object is provided by the vision system [20]. A Kinect v1 camera, next to the Baxter was used to observe human intent, and a Kinect v2 camera on top of the Baxter's head was used for the robot end of the architecture.

A joint tea-making task was designed based on the task tree which encodes the constraints of both THEN and AND nodes (Fig. 1). The scenario contained both overlapping and non-overlapping sub-tasks between human and robot.

The robot and the human both went for the cup to pick and place, which resulted in a collision. The robot started to negotiate; the human told the robot to finish the current task. While the robot was performing the task, the human moved to the next object, which was picking and placing the teapot. A collision

Fig. 4: A sample view of the experimental setup to perform a human-robot distributive collaborative tea table task



(a) Test Scenario: Baxter (b) Test Scenario: Human

Fig. 5: The timing diagrams of the tea-table task scenario on the human and the Baxter. These show the times at which the state of a node in a given task tree changed. Each row corresponds to a behavior node named as its corresponding object. The horizontal axis is increasing time. Brown → *inactive*, Orange → *active*, Green → *working*, and Blue → *done*.

was again detected as the human and the robot were both going for the apple which started the dialogue between the robot and the human again. The human wanted to perform the current task and informed the robot. The robot stopped going for the apple and moved to the next task to pick and place the burger.

5 Results

The timing diagrams (Fig. 5) illustrate the state for each node during scenario execution using the task structures of the human and robot shown in Fig. 1. There are four state types in the diagram: *inactive*, *active*, *working*, and *done*. Each state is shown with different color bars in the diagram for each node.

When the task starts, both the cup and teapot are eligible for both agents to grasp (due to the task tree constraints), thus becoming *active*. At first, both agents choose to go for the cup which caused a collision and began a dialogue. As in the task design, the human let the robot finish the task for this collision resulting in the cup status of the robot being changed to *working* (Fig. 5a). While the robot was finishing the task, the human moved on to pick and place the teapot, which changed the teapot node status for the human to *working* in Fig. 5b, due to the human's action. After placing the cup and the teapot, the status of both objects became *done* in both agents.

After the teapot and cup were completed, the apple and burger became eligible for grasping by both agents (due to the task tree constraints), and so their status became *active*. The second collision occurred on the apple task. After the Baxter began working on the apple task, the human started the same task, which triggered a collision and began a dialogue. The human told the robot to stop. The robot stopped working on the apple task (changing its state back to *active*) and moved on to the burger, changing its state to *working* (Fig. 5a).

Fig. 5b shows the human's apple node status changed to *working* (after the robot stopped working), as the human chose to finish the apple task. Once the apple was placed, the status was changed to *done* for both agents. Likewise, after the burger was finished by the robot, the status was set to *done* for both agents.

Based on the experiment, we see that our architecture is able to dynamically allocate tasks in a human-robot team. The system allows a human and robot to negotiate to resolve collisions that arise during the task allocation in order to complete the joint task. The subset of our architecture that this experiment validates shows that our system would be able to dynamically allocate tasks between a human partner and a robot partner that involve sequencing and multiple ordering of execution hierarchical constraints.

6 Conclusions and Future Work

This paper proposed a control architecture that performs a set of distributive collaborative tasks between a human and robot as a team. Tasks were performed by following a hierarchical representation which is responsive to a changing environment. This architecture has the following contributions:

(1) The robot maintains its own state and the state of its collaborative human partner. A human intention system, designed as an augmentation to our previous robot architecture, continuously publishes a message containing the human intention status information for each object.

(2) This allows for agents to operate independently when all agents are working on non-overlapping tasks; however, when agents' goals overlap, a collision occurs on the task tree, and dialogue is used to resolve the collision. This allows one agent to finish the task and the other to move to a different task.

The OR node functionality is not included in the task tree for task due to the complexity of collision resolution. A collision may occur if the human and the robot go for any of the objects that are children of an OR node at the same time. Thus, if the agents choose different children, it would be difficult to detect a collision and begin a dialogue for resolution. This functionality will be implemented in this architecture in the future which will allow for human-robot collaboration for tasks with alternative paths of execution. Again, the system isn't flexible enough to deal with the human error after the collision detection. In addition, the current architecture for collaborative tasks can be extended to a multi-human-robot architecture for a more robust collaboration.

Acknowledgment

The authors would like to acknowledge the financial support of this work by Office of Naval Research (ONR) award #N00014-16-1-2312,N00014-14-1-0776.

References

1. R. C. Arkin et al. An ethological and emotional basis for human–robot interaction. *Robotics and Autonomous Systems*, 42(3-4):191–201, Mar 2003.

2. S. Thrun et al. Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.
3. K. Severinson-Eklundh et al. Social and collaborative aspects of interaction with a service robot. *Robotics and Autonomous Systems*, 42(3-4):223–234, mar 2003.
4. M. Hans et al. robotic home assistant care-o-bot: past-present-future. IEEE.
5. R. Ambrose et al. Robonaut: NASA's space humanoid. *IEEE Intelligent Systems*, 15(4):57–63, jul 2000.
6. J. Khurshid and H. Bing-rong. Military robots - a glimpse from today and tomorrow. In *ICARCV Control, Automation, Robotics and Vision Conference*, 2004.
7. J. Casper and R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(3):367–385, jun 2003.
8. M. Awais and D. Henrich. Human-robot collaboration by intention recognition using probabilistic state machines. In *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*. IEEE, jun 2010.
9. Z. Wang et al. Recurrent convolutional networks based intention recognition for human-robot collaboration tasks. In *International Conference on Systems, Man, and Cybernetics (SMC)*, 2017.
10. Y. Li and S. S. Ge. Human–robot collaboration based on motion intention estimation. *IEEE/ASME Transactions on Mechatronics*, 19(3):1007–1014, jun 2014.
11. P. A. Lasota and J. A. Shah. Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 57(1):21–33, jan 2015.
12. R. C. Luo and C. Huang. Human-aware motion planning based on search and sampling approach. In *Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 2016.
13. S. E. Navarro et al. Methods for safe human-robot-interaction using capacitive tactile proximity sensors. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, November 2013.
14. D. Feil-Seifer and M. Matarić. People-aware navigation for goal-oriented behavior involving a human partner. In *Proceedings of the International Conference on Development and Learning (ICDL)*, Frankfurt am Main, Germany, August 2011.
15. J. Baraglia et al. Initiative in robot assistance during collaborative task execution. In *International Conference on Human-Robot Interaction (HRI)*, 2016.
16. K. P. Hawkins et al. Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 499–506, Oct 2013.
17. K. P. Hawkins et al. Anticipating human actions for collaboration in the presence of task and sensor uncertainty. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2215–2222, May 2014.
18. J. Blankenburg et al. A distributed control architecture for collaborative multi-robot task allocation. In *International Conference on Humanoid Robots*, Birmingham, UK, November 2017.
19. M. Quigley et al. Ros: an open-source robot operating system. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.
20. S. Pourya Hoseini A. et al. Handling ambiguous object recognition situations in a robotic environment via dynamic information fusion. In *Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, June 2018.