University of Nevada, Reno

Distributed Dynamic Hierarchical Task Assignment for Human-Robot Teams

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in
Computer Science and Engineering

by
Bashira Akter Anima

Dr. David Feil-Seifer/Dissertation Advisor

December 2022

THE GRADUATE SCHOOL

We recommend that the dissertation
prepared under our supervision by

BASHIRA AKTER ANIMA

entitled

Distributed Dynamic Hierarchical Task Assignment for Human-Robot Teams

be accepted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

David Feil-Seifer, Ph.D.
Advisor

Monica Nicolescu, Ph.D.
Committee Member

Christos Papachristos, Ph.D.
Committee Member

Alireza Tavakkoli, Ph.D.
Committee Member

Adam Kirn, Ph.D.
Graduate School Representative

Markus Kemmelmeier, Ph.D., Dean
Graduate School

December, 2022

Abstract

This work implements a joint task architecture for human-robot collaborative task execution using a hierarchical task planner. This architecture allowed humans and robots to work together as teammates in the same environment while following several task constraints. These constraints are 1) sequential order, 2) non-sequential, and 3) alternative execution constraints. Both the robot and the human are aware of each other's current state and allocate their next task based on the task tree. On-table tasks, such as setting up a tea table or playing a color sequence matching game, validate the task architecture. The robot will have an updated task representation of its human teammate's task. Using this knowledge, it is also able to continuously detect the human teammate's intention towards each sub-task and coordinate it with the teammate. While performing a joint task, there can be situations in which tasks overlap or do not overlap. We designed a dialogue-based conversation between humans and robots to resolve conflict in the case of overlapping tasks.

Evaluating the human-robot task architecture is the next concern after validating the task architecture. Trust and trustworthiness are some of the most critical metrics to explore. A study was conducted between humans and robots to create a homophily situation. Homophily means when a person feels biased towards another person because of having similarities in social ways. We conducted this study to determine whether humans can form a homophilic relationship with robots and whether there is a

connection between homophily and trust. We found a correlation between homophily and trust in human-robot interactions.

Furthermore, we designed a pipeline by which the robot learns a task by observing the human teammate's hand movement while conversing. The robot then constructs the tree by itself using a GA learning framework. Thus removing the need for manual specification by a programmer each time to revise or update the task tree which makes the architecture more flexible, realistic, efficient, and dynamic. Additionally, our architecture allows the robot to comprehend the context of a situation by conversing with a human teammate and observing the surroundings. The robot can find a link between the context of the situation and the surrounding objects by using the ontology approach and can perform the desired task accordingly. Therefore, we proposed a human-robot distributed joint task management architecture that addresses design, improvement, and evaluation under multiple constraints.

Acknowledgements

I would like to thank my committee members: Dr. David Feil-Seifer, Dr. Monica Nicolescu, Dr. Christos Papachristos, Dr. Alireza Tavakkoli, and Dr. Adam Kirn for their support, patience, and time. I would especially take this opportunity to thank Dr. David Feil-Seifer for being my advisor and giving me the opportunity to research in the Socially Assistive Robotics Group (SARG). I would like to acknowledge my lab-mates to help in my research: Dr. Janelle Blankenburg, Mariya Zagainova, Dr. S. Pourya Hoseini A., Muhammed Tawfiq Chowdhury, Roya Salek Shahrezaie, and Andrew Palmer.

I like to thank my friends here in Reno for being my family. I also thank my friends who are not physically here but continuously gave me support and inspiration to get through these tiresome times. Thanks to robot Baxter for existing during those frustrating times at the lab although it was the Baxter who was giving me the hard times most of the time!

Last but not the least, I like to thank my parents and my brother for their unconditional love and support throughout my life. This day wouldn't even be possible without my parents' love and sacrifices. I am dedicating my Ph.D. to my parents and brother who are currently thousands of miles away from here in Bangladesh.

This material is based in part upon work supported by: The Office of Naval Research under grant number(s) #N00014-16-1-2312 and #N00014-14-1-0776, National

Science Foundation (NSF) award #IIS-1719027, #IIS-2150394, and Army Research Laboratory (ARL) award #W911NF-20-2-0084. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the Office of Naval Research or the Army Research Laboratory.

Contents

List of Figures

Chapter 1

Introduction

The purpose of this work is to enable naturalistic human-robot task execution for complex hierarchical tasks, followed by evaluating the task structure. This task architecture is based on a complex hierarchical task network in which humans and robots can work simultaneously in the same environment. Three types of task constraints can occur in that situation: 1) sequential order 2) non-ordering, and 3) alternative execution constraints. Almost any real-world task can be divided into sub-tasks and designed as a hierarchical task network. Similar to a hierarchical task network, a real-world task may comprise sequential, non-sequential, and alternative sub-tasks. For example, we are focusing on on-table tasks such as organizing a tea table or assembling a table top which consists of sequential, non-sequential, and alternative sub-tasks. In order to complete a building task, some sub-tasks can be performed without maintaining any order, whereas some require an ordered sequence to finish.

To make a cup of tea, the sub-tasks can include (*) bringing a cup to a specific place on the table, (*) pouring tea, and (*) adding milk and sugar. In this case, adding milk and sugar to the cup does not require a specific order. However, bringing a cup to the table and then pouring tea should be done sequentially.

If multiple agents want to finish a task together as a team, then all of them should be able to allocate their sub-tasks while following the task structure. Individuals should be able to define their sub-tasks according to their constraints without restriction. There shouldn't be any predefined task allocation for each agent. As ideal teammates, both agents should have the freedom of choice to perform their sub-tasks based on the task's constraints. Additionally, teammates should also be able to know which part of the sub-task is already finished or is being performed by the other teammate so that they don't need to attempt to finish the same sub-task again. However, there may also be situations where both teammates are working on the same sub-task. To solve this issue, the teammates must reach an agreement where one of the teammates will continue working on the sub-task, and the other teammate should start working on the next sub-task according to a defined task structure.

To implement this idea, we developed our previous multi-robot domain architecture [2] into a multi-robot-human domain to make it a collaborative robot-human task structure. In order to make this change, it was necessary to design the robot's decision-making process to choose the sub-task to perform. For this, the robot needs to monitor the status of the sub-tasks continuously. Each agent needs to plan based on the actions of their teammates while following the task constraints. All these actions are combined into joint actions. If an agent wants to execute an action, it requires information about the activities and goals of peers as well as knowledge about the state of the environment. Based on this knowledge, the agent can take a proper decision to fulfill the joint plan while not obstructing the other peer's sub-task goal. When multiple agents' sub-goals may overlap, a re-plan solution should be designed while ensuring a smooth collaborative process.

1.1 Multi Human-Robot Collaboration

In the case of robot-robot collaboration, both agents communicate using a distributed message-passing system that holds the information of the team member's states. This allows them to keep track of which part of the task model their peer teammate is working on or has already executed. As a result, the robot was able to decide which remaining sub-task needed to be finished while not blocking other teammates' sub-task goals. However, in the case of human-robot collaboration designing a direct distributive message-passing system between the human and the robot can be quite challenging. The system needs to understand the human teammate's goals and intentions to be able to send the correct state information through the message-passing system. Understanding each other's intentions will enable them to work together to achieve the goal.

Communication of intentions can take the form of speech, gestures, actions, etc. Some of these intention techniques include explicit ways of communication such as using speech, hand or head gestures, eye gaze, and pointing gestures. However, a teammate may not want to communicate in such a way for every sub-task. Usually, teammates don't state their intentions explicitly during a human-to-human collaboration. Peers estimate each other's intentions by observing their movements. By applying this idea, human intention can also be derived in this way for a robot-human collaborative task. A robot should be able to recognize the intention of the human peer in that case by monitoring the activities of the peer. By doing this, the human teammate won't be interrupted while working on the joint task.

Furthermore, there can also be some situations where the robot may perceive that both of them are heading for the same sub-task. If we again think about a human-human collaboration, the human teammates will try to resolve this situation by letting one of them finish the task while the other teammate will proceed to the next task based on the task model. Humans and robots can also communicate to resolve situations using human-robot task design. We believe that by doing this the human teammate will have the flavor of working with a teammate.

Therefore, based on this idea, in our proposed human-robot hierarchical task architecture humans and robots can perform sequential, non-sequential, and alternative tasks altogether. Both of the teammates will be aware of the update of each sub-task and will be aware of which teammate was performing which sub-task. The robot teammate will continuously be updated about the peer teammate's updates of each sub-task. Furthermore, the robot will detect intention by observing the hand movements of its human partners and to aid its decision-making. Thus, the robot will be able to perceive whether they are aiming for the same sub-task or not. If this situation occurs, our architecture will offer a resolution by holding a conversation in natural language between the robot and human teammates. The robot and the human will decide on who is responsible for which sub-task in the conversation.

1.2 Trustworthiness

In designing a collaborative human-robot task architecture, evaluating the design is one of the primary concerns. To evaluate the performance of a system, many metrics can be measured. One of the vital metrics is the trust and trustworthiness of the system.

Human-robot trust has become a crucial part of today's world when there is a rise of social robots in many fields. Therefore, evaluating a human-robot task model by measuring metrics such as trustworthiness towards the robot has become a very essential aspect. Trustworthiness is the property of an agent's vulnerability and confidence towards the other agent. [3]. A trustworthy robot is one that can evoke trust in its interaction partners with its actions. Trust is a very significant metric for designing autonomous and semi-autonomous technologies, because "No trust, no use"

[4]. One of the main goals of social robotics is to develop trustworthy relationships. It was even said that " No matter how capable an autonomous system is if human operators do not trust the system, they will not use it". [5].

In times of human-human interaction, we ask ourselves "how trustworthy is the other person"? When we meet a person for the first time, we find out if we trust them based on different events or property. It can be the person's physical appearance, intentions, goals, values, origin, similarities, posture, facial expression, body movement, gaze, and so on. Users may also be able to trust a robot if these properties are present. A robot's physical appearance can create a positive first impression for the human user. If a robot tends to show more human-like characteristics, it becomes more trustworthy from the human's perspective [6].

Along with physical appearances, how a robot socially interacts while collaborating with human users or handling a task can also affect trustworthiness. For instance, if we meet another person who shows similarities through their physical actions we may form a bias towards them. This phenomenon is called homophily. Similarly, people may form a connection with the robot if the robot shows human-like emotion or personal understanding via its activities.

Usage of natural language to communicate with the human user or explain its own actions can play an instrumental role to create trustworthiness. It also helps to mitigate the loss of trust due to the robot's performance [7].

To understand if this will affect trust, we have also designed interaction between human and humanoid robots [8]. In our proposed design, the robot was trying to talk about a topic that the human would like. This interaction was intended to establish homophily between them. As a first step, we determined if humans and robots could form a homophilic relationship. Later, we explored the correlation between homophily and trust in our human-robot interaction. We used questionnaires and human participants' feedback to understand and measure trust and homophily in human-robot interaction.

1.3 Task Demonstration and Learning

The aim of designing a human-robot collaborative task architecture is to make it efficient, flexible, and dynamic. If robots are used in an environment with humans to perform tasks together, then some special abilities or features are anticipated from the robots. As an example, humans and robots will have to interact to accomplish complicated tasks in the same way as human-human teamwork. However, the situation in human-robot domains is different. It is easy for one human teammate to teach another human teammate about the task or to provide input to a revised task design. Information can be conveyed in many ways, including speaking, showing, gesturing, or gazing. However, usually, in order to learn a new task design, the robots need to be manually specified by the programmer. Additionally, if something in the task design needs to be updated again the robot needs to be programmed manually each time. These situations are not realistic for teamwork situations. Because not every teammate in the domain will be proficient in robot programming.

Due to this reason, if the robot can directly learn from the teammate or human teacher without being manually defined by the programmer, then the system will become more efficient and usable. The robot can learn and update the new task by monitoring the human teacher's movement. A vision system can be applied for the robot to monitor the human teacher's hand movement and follow the task design. After observing the task, the robot should be able to create and execute the hierarchical task tree in its system.

A dialog-based conversation based on natural language can also be used to teach the robot about task design. As it was already mentioned before, using natural language to communicate between humans and robots would also increase trustworthiness. If the robot can learn about the situation in an instant and can decide what task it requires to perform then the human teammate need not teach the robot each task step by step. For example, if the robot was told that someone is thirsty then the robot can observe the environment and search for the ingredients to resolve the situation. After that, with the appropriate objects, the robot should be able to create a design to perform the task.

We thus proposed a system that will learn from human demonstrations by monitoring the movement of human teammates and having dialogue conversations with them. The robot will be able to assign tasks on its own when it creates the hierarchical task tree from the human demonstration. This will reduce the need for a programmer to constantly feed the robot information. Additionally, the robot can also understand the context of a dialogue-based conversation and decide what type of task it is required to do.

1.4 Contributions

Accomplished contributions, directly related to this dissertation are listed below:

- **Collaborative Human-Robot Hierarchical Task Execution**: Developing our previously designed hierarchical task execution model from multi-robot team to multi-human-robot team where the tasks with different environmental conditions are given dynamically. In order to do this, we extended the system's ability to accommodate and anticipate a human's movement. The human-robot collaborative task is represented in a tree structure that consists of sequential, non-ordering, and alternative paths of execution. 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 robot decides its next action based both on the constraints of the defined task and the behavior of the human partner.
  - The robot monitors 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.
  - 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.

- **Study on homophily and trust in HRI**: Explored homophily between a person and a robot by measuring metrics such as common interest, bonding, and similarity.
  - The purpose of this work is to determine whether similarities between a robot and a person might improve social connection and trust. If such a link exists, then homophily would be an important physical and behavioral design consideration for effective HRI.
  - Measured "common interest", "felt bonding", and "trust" between homophilic and non-homophilic conditions.

- **Efficient task allocation and execution from task demonstration:** The robot is able to learn a new task model by monitoring the human teacher's movement. This reduced the work on feeding the hierarchical task tree to the system manually for different task designs before running the system. As a result, the system became more efficient, flexible, and dynamic.

  - Designing and implementing a pipeline where the robot will observe the task demonstrated by the human teacher. It will learn the task design by using the previously proposed learning framework and execute the new task tree by itself.
  - Designing a task demonstration interface by using a vision based system to learn the task demonstrations from the human teacher. The robot will observe the human teacher performing the task while conversing with the human.
  - Enhance the system's ability to learn and execute new tasks without manual specifications.
  - Demonstrating this on a real robot.

- **Cognitive Approach to Hierarchical Task Selection For Human-Robot Interaction in Dynamic Environments:** An efficient and flexible human-robot collaboration environment is designed in which the robot teammate can perform the user's desired task by deciphering both vague or clear requests in a natural language form from the human teammate.
  - Finding an implied link between the context of the situation and the surrounding environment using the ontology approach after interacting with the human user.

- In our extended hierarchical task architecture, the robot will only select the hierarchical sub-tasks that are most relevant to the specific task derived from the ontology approach.

.

1.5 Summary

This work aims to improve the coordination of a joint task architecture in which humans and robots can collaborate as teammates. This chapter looked at the complex hierarchical task architecture representing a complex real-world task, such as a building project. In this task scenario, robots and humans can collaboratively perform sequences of sub-tasks sequentially, non-sequentially, and alternatively. In addition, the system allows all teammates to allocate sub-tasks while staying informed of their peers' status based on the task definition and constraints. We also discussed the impact of trustworthiness in human-robot interactions and our plan for task demonstration. Later, we presented our proposed contributions in detail.

Chapter 2 addresses the idea of hierarchical task networks and the advantages of using HTN planning in robotics applications. Later, we talked about various HTN applications in a multi-agent system. Moreover, we discussed using the AND/OR/THEN structure for hierarchical task networking to represent different sequential, non-sequential, and alternative constraints. Following that, we discussed various research on collaborative human-robot hierarchical tasks, human-robot interaction design, and task demonstration interfaces. Furthermore, we also explored multiple works on trust and trustworthiness. Chapter 3 discusses our previously developed multi-robot control architecture where multiple agents have a distributive message passing system between them to publish their current states. This mechanism helps all the robot teammates define their next plan to finish the task without hindering other agents' work. In chapter 4, we explain our proposed collaborative human-robot hierarchical task architecture. In this work, we extended the previously developed multi-robot architecture to multi-human-robot architecture where humans and robots can work as teammates under the same environmental conditions. In this system, the robot can detect a human teammate's intention and decide which sub-task to focus on without hindering the other teammate's sub-task goal. However, there can be situations when the human and robot teammates attempt to complete the same sub-task. In that situation, the system triggered a dialogue-based conversation between the teammates to resolve. Chapter 5 focuses on the importance of trust and the connection between trust and homophily in Human-robot interaction. We designed interaction between humans and robots to create homophily between them to determine the validity of our proposed hypotheses regarding homophily and trust. The experiment showed a promising result, and we found out that there is a correlation between homophily and trust. After realizing the importance of trust, we wanted to evaluate the performance of our proposed collaborative human-robot task architecture by measuring metrics like trustworthiness, collaboration style, comfort, and so on. In chapter 6, we present an efficient, flexible, and dynamic human-robot collaborative task architecture. We proposed a pipeline where the robot can directly learn from a human teammate to learn or update a new task without being programmed by a programmer. In our proposed task demonstration interface, the robot can monitor the human teammate's movement while conversing with them in natural language to observe the upcoming task. Afterward, the robot can learn the task utilizing a genetic algorithm-based framework and execute the task in real time. Chapter 7 proposes a solution where the robot can understand the context of the environment while being in a human-robot collaborative environment. Holding a dialogue conversation with a human teammate can allow the robot to understand the context of the environment by deciphering both vague and explicit requests. After receiving a response from the human end, the robot uses ontology to determine which task it should perform, considering the objects in the environment. This process eliminates the need to explain the situation to the robot explicitly, mimicking a more human-human interaction in human-robot teams. In Chapter 7, we discussed the summary of our dissertation. We also discussed its limitations and future work on it. We also provided publications that have already been published and submitted as a result of our dissertation research.

Chapter 2

Related Work

This chapter briefly discusses previous works in hierarchical task network planning and its use in a multi-agent system. We talked about why HTN can be a suitable choice for our collaborative human-robot task architecture. Additionally, we will discuss the importance of trustworthiness in designing social robots. We found several previous papers exploring this topic and identifying the phenomena that create trust toward robots. We also briefly describe different types of human-robot interaction designs in various research and the reason for picking our interaction style. Also, we present some prior works on teaching a robot a revised task through a task demonstration interface.

2.1 Hierarchical Task Network Planning

The idea of a hierarchical structure plays an imperative role in understanding and conceptualizing the world. In the field of Artificial Language (AI), a hierarchical task network (HTN) is a technique to perform automated planning which is different from classical planning [9]. In order to do this, dependency among actions is provided in the form of domain-specific hierarchically structured task networks with primitive and compound tasks. To solve an HTN planning from the initial task network, compound tasks are decomposed into a set of simpler primitive tasks with ordered constraints.[10]

An HTN planner helps to divide complex behavior into simpler task behaviors and allows easier user integration in the plan generation process. Its ability to use domain-specific problem-solving knowledge can improve the planner's performance dramatically and sometimes make the difference between solving a problem in exponential time and solving it in polynomial time [11]. Additionally, experiments showed that Hierarchical Task Network (HTN) planners are suitable to find solutions for nontrivial tasks in complex scenarios [12].

It was observed that HTN planning was useful for robotics applications because of its domain-specific knowledge structure [13]. Instructions from the domain expert in the domain are presented as an intuitive hierarchy. This helps guide the search and as a result, it makes the system faster in general than classical planning approaches. Therefore, it is more practical for real robots which require them to be responsive in case of environmental changes.

2.2 HTN in multi-agent system

Mostly, the HTN method in robotics focuses on single robot navigation [14] [15] and task planning [12]. Because of this lack of consideration towards multi-agent space applications, a domain configurable autonomous multi-agent space system [16] and a multi-agent extension of the HTN planning formalism [17] were proposed. Moreover, in [18], the HTN method is applied to multi-robot path planning by searching for an optimal or approximate optimal collision-free path from start to goal state.

HATP (Hierarchical Agent-based Task Planner) planning framework was proposed to make HTN more suitable for the robotics field by extending the traditional HTN planning domain representation [19]. They were concerned more about social interactions between humans and robots which is a big challenge [20]. By socially interactive robots, it's stated in [21] that they must "operate as partners, peers or assistants, which means that they need to exhibit a certain degree of adaptability and flexibility to drive the interaction with a wide range of humans".

A combination of hierarchical task network planning with modern constraint reasoning techniques was used to develop MACBETH [22]. It has been tested on Unmanned Combat Aerial Vehicles (UCAV) sorties and Tactical Mobile Robotics. It contains constraints AND/OR tree search hierarchy. Human users can specify instructions and constraints on tasks via a playbook GUI. Using the AND/OR graph representation of assembly plans allows the selection of the optimal assembly plan, recovery from execution errors, and opportunistic scheduling. An AND/OR graph offers parallel execution of assembly operations and time independence operations can be executed in parallel [16].

Because of the above-mentioned reasons, we used a Hierarchical Task Network representation using AND/OR graph structure for our distributive collaborative architecture. AND/OR graph representation offered to show the task model with various constraints such as ordering, non-ordering, and alternative path of executions.

2.3 Collaborative human-robot Hierarchical Task

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 [23]; museum documents [24]; personal assistants [25]; health care [26]; space exploration [27]; police SWAT teams [28]; military robotics [28]; and rescue robotics [29]. 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 [30]. Recurrent

Convolutional Neural Networks (RCNNs) [31] and Neural networks [32] were used to detect human intention, and an online estimation method was developed to deal with the nonlinear and time-varying properties of a limb model. Human-aware motion planning was examined in [33] and [34]. The ability of a robot to work with a human in close proximity [35] without colliding with the human was demonstrated. A Gaussian Mixture Model (GMM) representation [36] 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. However, our work focuses specifically on the detection of human intention by robots while much of the above prior work is concentrated mainly on robot awareness of a person.

Given detected intent, it is an open question whether and when a robot should take initiative during joint human-robot task execution [37]. 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. There have been researches on empathy-generating robots [38] that can petition on their own behalf or someone else's behalf to avoid penalties. Their experiments attempted to achieve a delicate observation of empathetic motivation by exploring humans' reactions to an empathy-inducing robot. That work focuses primarily on human's observation of robot's intentions and artificial feelings. Handover of objects between human and robot [39], [40] were examined and [39] looked into the concept of robot's eye gaze as a medium of communication in cases where communication via speech is not feasible. How robots are designed to involve in physical collaborations may achieve similar adaptivity in performing handovers is observed in [40] and the implemented autonomous system was assessed in a human-robot interaction study against two baselines that use "proactive" and "reactive" coordination procedures. They performed their experiments in the household scenario. They collected data from pairs of human participants as they performed handover actions under different task demands. The analysis of this data resulted in a computational model of adaptive coordination that was implemented on a robotic manipulator.

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 [41]. 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 [42], 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. As a result, we were able to design a distributive message-passing system between the human and the robot peers which allows the robot to know the current situation of the task tree.

2.4 Human-robot task interaction design

In order to perform a collaborative task together, all the teammates are required to share their joint tasks. These tasks can be assembling tasks or completing construction tasks. In the case of human-robot collaboration, robot behavior needs to be designed in such a way that will help the human-robot team to coordinate their tasks and improve their task performance [43]. Task planners based on HTN planners were used in the work of Gregoire, et al., [44] and Sandra, et al., [45] to adapt collaborative plan generation and coordination.

In order to share joint actions, team members need to communicate and interact with each other to successfully achieve the task goal. To gain effective coordination, joint attention, action observation, sharing tasks and task coordination play an important role [46].

Interaction style, or how a robot interacts with the human with respect to autonomous action or command-driven action, can also affect the efficiency of interactions and perceptions about the robot [43]. Regarding on-table tasks between humans and robots, collaborations can be of different types of interaction styles, such as human-initiated, robot-initiated reactive, and robot-initiated proactive [47]. Human-initiated means the robot will help the human teammate when they will ask for it specifically. Robot-initiated reactive means the robot will help when it detects that help is required. And, robot-initiated proactive means the robot will perform possible actions while keeping human actions in mind. It was observed that humans preferred proactive and autonomous modes while performing a task together instead of human-requested or human-commands mode for each step [47] [48]. Therefore, in our proposed work the human and the robot will perform the on-table task together in a proactive mode where both of them will follow similar constraints to reach the goal.

2.5 Homophily in HRI

In a social group, it is observed that there is a tendency for similar people to be connected. This phenomenon is called homophily which means love of the same [49]. People's values also can be affected by other similarly-minded people's opinions.

According to the homophily effect, similar users are more likely to establish trust relations. [50]. When it comes to making a decision, people with similar likes tend to trust each other. By discussing some previous works on homophily, we will try to determine whether we can establish this homophily in human-robot interactions and the consequences.

Homophily is a term familiar in social sciences. In Rhetoric and Nichomachean Ethics, Aristotle noted that people "love those who are like themselves" [51]. It was also observed by Plato that "similarity begets friendship" [52]. McPherson, et al., [49], presented a principle named homophily. It states that "a contract between similar people occurs at a higher rate than among dissimilar people." Overall homophily can be differentiated into two types: 1) value homophily and 2) status homophily. Value homophily is based on attitudes, beliefs, and values. Status homophily is based on national origin, sex, age, and characteristics like religion, education, and occupation.

Much research in the robotic world also worked on the common factors that a robot and a human can share. In one study, Jung, et al.[53] presented the preferences of humans and robots regarding different aspects of human and robot interactions based on their characteristics and facial expressions. Two types of personalities: extrovert and introvert were applied to the robot named KMC-EXPR to observe the impact of different personality types in the interaction between humans and robots. Also, in Kahn's work [54], a humanoid robot named Robovie was used to interact with children. After each 15 min session, the experimenter interrupted the session and sent the robot to the closet. Later, it was observed how the children felt toward the robot in many aspects.

The effect of verbal and nonverbal behavior based on personality traits in human-robot interaction has been observed [55]. An NAO robot was used to validate their model that a person preferred more robots to interact with if they both had the same personality traits. Finally, a study from Heerink [56], shows that age, gender, education, and computer experience had an influence on robot acceptance by older adults. Our prior work showed that establishing common ground using ice-breaker tasks helped a person identify with a robot team-member [57]. Witnessing verbal mistreatment of a robot also resulted in an increased perception of the robot's emotional ability [58].

Recent work investigated if a human user would help a robot being bullied by other humans when social bonding has been applied in human-robot's interactions [59]. Similar to our study, they used favorite food to contextualize a human and robot conversation so the person finds a similarity with the robot. Their results did not prove their hypothesis, on the other hand, our findings suggest that a shared similarity can improve sympathy in human and robot interaction.

2.6 Trust in HRI

It is observed that people tend to trust more easily those people who appear similar to themselves. By similarity, it may include common values, membership in a defined group (such as manufacturing departments, a local church, gender), shared personality traits, etc. [60]. In that research, when people evaluate others' trustworthiness, cues such as gender [61], age [62], race, and nationality influence the initial assessment.

Salem et al. [63], conducted an experiment in which participants interacted with a home companion robot in one of two experimental conditions named correct mode and faulty mode while tapping different dimensions of trust based on a variety of unusual collaborative tasks. It was observed that the robot's performance did not influence participants' decisions to comply with its request. Hancock et al. evaluated the effects of the human, robot, and environmental factors on perceived trust in human-robot interaction [64]. Human-related factors depend on ability-based characteristics, robot-related factors are based on performance and attributes, and environmental factors include team collaboration and tasking. In this study, [65], whether a robot's vulnerable behavior can create ripple effects on a team and increase team physiological safety and human-human trust-related behavior was explored. It was seen that the 'ripples' of the robot's vulnerable behavior influence not only team members' interaction with the robot but also team members' human-human-trust-related interaction with each other.

Based on these works, we explored a relationship between homophily and trust in human-robot interaction in our proposed work. The result showed us that homophily creates an impact on building trust. As a result, we are exploring the trustworthiness of a robot further in our system architecture. The effectiveness of our system architecture can be evaluated by observing whether human teammates perceive robot teammates as more trustworthy and the impact this has on performing collaborative tasks.

2.7 Task Demonstration Interface

The methodology of learning from demonstration (LfD) is an approach that enables robots to perform new tasks by themselves. Instead of requiring users to decompose the desired behavior and program it manually, LfD believes that a robot controller can be derived from observations of a human doing it manually. We aim to enable robot capabilities to be easily extended and adapted to novel situations, even by non-programmers.

Demonstrating a task can be done in several ways, and the interface is instrumental in delivering the demonstration to the robot. It can be using a vision-based motion tracking system, Kinesthetic teaching, or Immersive teleoperation scenarios. The recording of human motion using vision can be very useful for teaching a new movement task and has been used in a variety of works [66][67][68]. In this technique, humans can move freely to teach the robot. However, if the robot's joint action capabilities are very different from the human, it can be challenging to learn these tasks properly.

By kinesthetic teaching, the robot is physically guided through the task by humans. This process enhances the ability to teach a task naturally and accurately. Also, there is more chance to fix a skill. However, if the task is a synchronization task that may need multiple limbs, then it is not easy to teach it through this process.

In an immersive teleoperation scenario, the human teacher needs to rely on the robot's sensors and has to instruct using these. For example, teleoperation to teach a task can be performed using a remote control device that allows the human teacher to keep a distance between themselves and the robot. In the work of Abbeel, et al.[69], an expert pilot teleoperated a helicopter to learn the acrobatic trajectories by recording the tilt and pan motion of the helicopter. A human teacher taught a robot dog to play soccer by guiding it via a joystick in the work of Grollman and Jenkins [70]. However, in this process, the human teacher needs to be familiar with the learning devices and efficient with them.

Because of this, we are looking forward to using the vision-based motion tracking approach to learn human demonstration. Having worked with humanoid robots, it will be easier for the robot's controller to comprehend and imitate the tasks. In addition, the human teacher is not required to be efficient in using learning devices such as a joystick.

In addition, verbal cues can be added to the vision-based system so that it understands the environment better.

Task structure learning focuses on learning the underlying structure of a given task. These types of methods focus on what steps need to be completed as well as the constraints inherent between the steps. For example, in a building task, the method must identify the order in which the parts need to be moved to correctly build the given structure while adhering to ordering constraints such as placing base blocks down before the roof. In other words, these methods focus on a form of task allocation in which the task structure is being learned.

Task allocation methods can use an auction scheduling algorithm [71, 72]. Some work extends this auction type of scheduling to allow task allocation on multi-robot teams [73, 74]. These methods focus on learning how to allocate sub-tasks to robots in order to complete the overall tasks. These methods focus only on learning a sequential ordering of tasks, which means they are limited in the tasks they can learn. For instance, they cannot deal with multiple choices within a task such as in cases where only sub-task A or sub-task B need to be completed but not both. Our proposed method is able to learn not only the *task ordering* but also a set of *task constraints*, such as alternate paths of execution.

Many methods, like ours, focus on hierarchical task representations to account for the constraints of a task. A common method is a hierarchical task network (HTN). Although many HTN methods are able to handle more diverse constraints such as alternate paths of execution, the scope of the works is different than our proposed method. Our proposed system is able to learn the components of a task from human movement. The methods in [75] and [76] focus on a question-asking module. In [77], policy exploration in a graph is used to build the HTN. The method in [78] uses a logic-based system to build their representation. The method in [79] focuses on natural language and scene navigation. The representation in [80] learns a graph with all paths through the task, whereas our method enforces a compact hierarchical representation. HTNs are also used in a wide variety of tasks outside of robot demonstration such as in storyline development [81] and orchestrating construction services [82].

2.8 Generalized Task Structure Learning

Task structure learning focuses on learning the underlying structure of a given task. These types of methods focus on what steps need to be completed as well as the constraints inherent between the steps. For example, in a building task, the method must identify the order in which the parts need to be moved to correctly build the given structure while adhering to ordering constraints such as placing base blocks down before the roof. In other words, these methods focus on a form of task allocation in which the task structure is being learned.

Task allocation methods can use an auction scheduling algorithm [71, 72]. Some work extends this auction type of scheduling to allow task allocation on multi-robot teams

[73, 74]. These methods focus on learning how to allocate sub-tasks to robots in order to complete the overall tasks. These methods focus only on learning a sequential ordering of tasks, which means they are limited in the tasks they can learn. For instance, they cannot deal with multiple choices within a task such as in cases where only sub-task A or sub-task B need to be completed but not both. Our proposed method is able to learn not only the *task ordering* but also a set of *task constraints*, such as alternate paths of execution.

Many methods, like ours, focus on hierarchical task representations to account for the constraints of a task. A common method is a hierarchical task network (HTN). Although many HTN methods are able to handle more diverse constraints such as alternate paths of execution, the scope of the works is different than our proposed method. Our proposed system is able to learn the components of a task from human movement. The methods in [75] and [76] focus on a question-asking module. In [77], policy exploration in a graph is used to build the HTN. The method in [78] uses a logic-based system to build their representation. The method in [79] focuses on natural language and scene navigation. The representation in [80] learns a graph with all paths through the task, whereas our method enforces a compact hierarchical representation. HTNs are also used in a wide variety of tasks outside of robot demonstration such as in storyline development [81] and orchestrating construction services [82].

2.9 Cognitive Approach For Human-Robot Interaction

During Human-Robot-Interaction (HRI) humans (teammates) can work effectively with one or more robots if all the members utilize similar task representations [83]. If robots are used in an environment with humans to perform tasks together, then communication is likely needed between users and robots for effective collaboration [84]. Communication could be based on the explicit exchange of information. For example, if the robot is told that "pick me a red bottle" then the robot can observe the environment, search for the object, and carry out the specific task to solve the problem [1].

Explicit cues can be used to teach or create tasks, such as route planning [85], human navigation guide [86], learning [87], and execution of tasks [88]. In another contribution, an interactive approach through verbal command was applied to enable the users to teach tasks to a mobile service robot [87]. Nicolescu investigated robot task learning from language-based instructions and proposed a novel approach [89]. Context-appropriate rules were selected using context recognition, object detection, and scene data for socially aware navigation in public places [90].

In order to understand the environment and verbal cues from a human teammate, there is a need to develop associations between objects, their effects, and actions carried out by robots [91]. In addition to verbal cues, anthologies have been used to develop a relationship between objects and their properties [1, 92, 93]. Although this improved the HRI experience a bit limited relationship types (i.e. isA, hasA, prop, usedFor, on, linked-to, and homonym) were unable to extract the information from implicit cues [1, 92]. Ontology in the form of semantic memory was also being reported [93, 94] and addressed the cues like “make a sandwich” but was unable to process situations such as “I am feeling hungry”, in which the robot understands that there is a need to make the sandwich.

In this work, we are using semantic memory developed from WordNet and ConceptNet for the understanding of explicit cues by evaluating the similarity score between atoms of verbal commands and available objects. As a baseline control structure, we adopted Nature-inspired Humanoid Cognitive Architecture for Self-awareness and Consciousness (NiHA) [1] and induced hierarchical control architecture [95, 96] as part of procedural memory. Our previous hierarchical architecture [95, 96] involved humans and robots executing the entire tree to accomplish a specific function. Using this previous work as a basis, multiple tasks were represented as “skills” in the tree. Upon receiving the highest similarity score among the available task objects, the architecture performs the skill associated with that object.

2.10 Summary

In this chapter, we discussed hierarchical task network planning and its use in several multi-agent systems. In this type of planning, compound tasks are decomposed into a set of simple tasks with constraints. It helps to describe complex sets of behavior into simple task behaviors. It was observed that HTN planning was helpful in robotics applications because of its domain-specific structure. We briefly discuss constraints such as AND/OR/THEN tree search hierarchy in previous works.

Moreover, the effect of making the multi-agent hierarchical task tree into a human-robot collaborative design was also described here. In previous works, several approaches to intent recognition have been used for robot collaboration with humans. Additionally, we briefly mentioned how our use of the intention system could help our AND-OR-THEN task network choose actions without enumerating all possible paths. Later, different types of human-robot task interaction designs were discussed. The efficiency of the system is also impacted based on human-robot interaction. Previous works found three types of interactions: 1) human-initiated, 2) robot-initiated, and 3) robot-initiated proactive. According to some research, human users are more likely to use robot-initiated proactive interactions when performing a collaborative task. In this mode, the robot will perform possible actions while keeping human actions in mind. Because of this, we have also been using this type of interaction in our system.

Several studies have demonstrated the importance of trust and trustworthiness in robotics applications. We further discussed several types of properties or events that can affect trust toward a social robot. It can be physical appearances, intentions, goals, values, similarities, attitudes, and so on. Additionally, we also described homophily and its effect on human-human and human-robot interactions. Several previous studies have examined how humans feel about the common factors between humans and robots. Based on this concept, the significance of trust toward the robot from a human's point of view was also more precise.

In the end, several types of task demonstration interfaces and their usage were discussed. We talked about some work on various interfaces to teach the robot about the task without using robotic programming knowledge. Moreover, the importance of verbal cues with a vision-based interface was also briefly discussed. Based on these previous works, it is believed that using the semantic memory representation generated by verbal cues will significantly facilitate the robot's design of hierarchical task trees.

Chapter 3

Prior Work

3.1 Hierarchical Task Representation

In prior work, a hierarchical robot control architecture was proposed that enabled the system to encode tasks involving various types of constraints such as sequential, non-ordering, and alternative paths of execution [2]. *THEN* nodes represent sequential constraints, the *AND* represents non-ordering constraints, and the *OR* represents alternative paths of execution. This representation serves both as an encoding of the task constraints as well as the actual controller that is executed by the robot, as described in [2]. An example task for arranging a tea table scenario is shown in Fig. 3.1. 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.

In this architecture, there are two types of nodes which are 1) Goal Nodes and 2) Behavior Nodes. *THEN*, *AND* and *OR* nodes are under the Goal Nodes. The leaf nodes in the task tree structure are called the Behavior Nodes which encode the physical behaviors that the robot can perform.

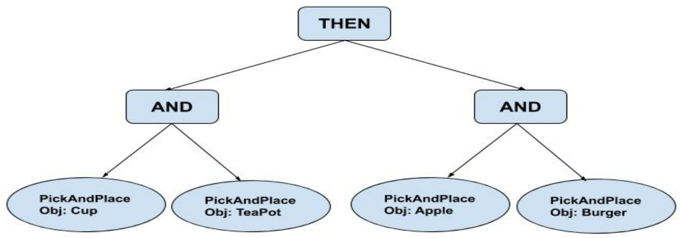


FIGURE 3.1: Hierarchical Task Representation

In order to execute a controller represented by such a hierarchical task, each node in the architecture maintains a state consisting of several components. They are:

- **Activation Level:** a number provided by the node's parent and represents the priority placed on executing and finalizing a given node
- **Activation Potential:** a number representing the node's perceived efficiency, which is sent to the parent of the node
- **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
- **Done:** a boolean variable that is set to true when the node has completed its required work.

The above state information 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.

As part of the task execution process, *activation spreading messages* are sent from the root of a task to its children to ensure that *activation level* is distributed throughout the task tree. Every node transmits its current state to its parent node as a *status messages* to propagate the *activation potential* throughout the tree in a bottom-up manner. Each cycle, an update loop is run, which helps keep the state of each node in the task structure up to date. The loop periodically checks the node's state and updates the various components of the state accordingly.

3.2 Multi-Robot Architecture

The single robot controller architecture was extended to a multi-robot architecture. It supports multiple robots by maintaining a copy of the task tree for each robot identical to other robots. In this scenario, the nodes that are equivalent across the task tree for each robot are called the peers. Peer nodes help the robots to keep track of each other. Message passing between peer nodes on the task tree allows each agent to represent the complete task status, not just the view from anyone agent.

Along with the single robot architecture components, two other elements were added to the multi-robot architecture. They are:

- **peer_active:** a boolean variable that is set true when the node or the node's peer node is active.
- **peer_done:** a boolean variable that is set true when the node or the node's peer node is done.

In addition to notifying when a robot is currently working on a behavior, it can inform when it has completed one and its activation potential and level. By this process, the robot can keep track of which task is completed and is being performed by the teammate and which task is required to achieve in the future. The robot can decide the next task to finish without hampering other teammates' work with this information.

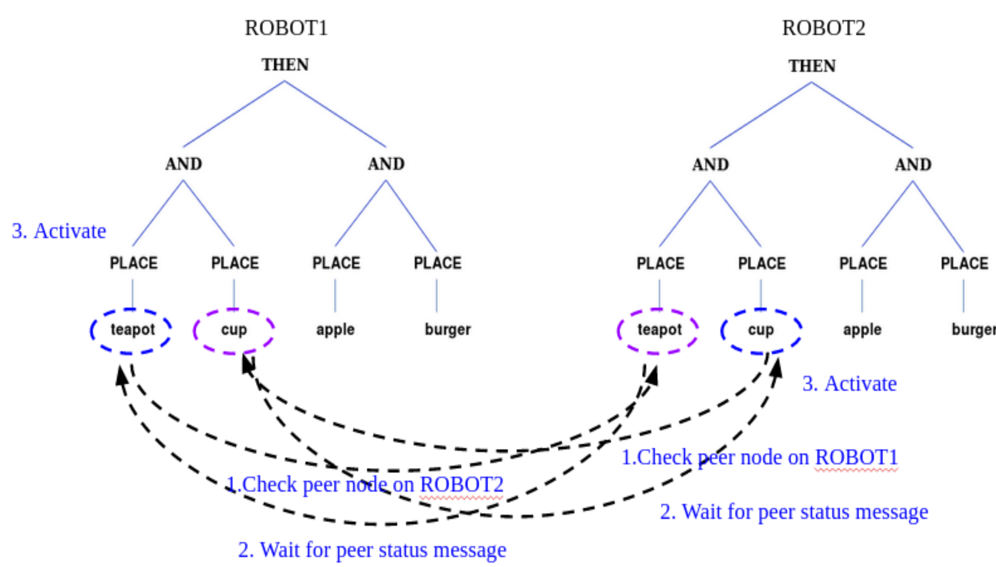


FIGURE 3.2: Multi-Robot Architecture

In Figure , we can see that to perform the PLACE-teapot task, the ROBOT1 will first check and wait for the status of its peer node teapot in ROBOT2. After getting the status message from the ROBOT2, if it finds out that ROBOT2 is not activating the same node, then ROBOT1 will start its node and perform the PLACE-teapot task.

Again, ROBOT2 will check and wait for the peer status in ROBOT1 to work on the PLACE-cup task. If it finds out that ROBOT1 is not intending to work on that task, then ROBOT2 will start working PLACE-cup task.

3.3 Human-Robot Collaboration and Dialogue for Fault Recovery on Hierarchical Tasks

The multi-robot architecture Incorporated a dialogue-based management system of task faults that can detect issues autonomously and resolve them by human-robot collaboration. For the fault detection system, a checking mechanism was added to the system. A Robotic Operating System (ROS) message is published to the corresponding node's issue topic if the system detects a fault.

There can be various types of issue messages to describe the fault:

- **Missed:** The missed issue message occurs when the robot misses an object to pick. The robot explains the situation to the human and asks to try again. If the human agrees, the robot will request the human place the object in its original position on the table and says it will try picking it up later. If the human disagrees, the robot will ask if the human will place the object. If the human replies with yes, they will place it in the final position. Otherwise, the robot will try again later.

- **Dropped:** This issue message occurs when the robot drops an object after picking it up. The dialogue will be the same as in the Missed issue case, except the robot will explain it dropped the object instead of missing it.

- **Unreachable:** This issue message occurs if a robot is not able to reach an object. The robot will ask for assistance from the human by asking if the human can hand the object to the robot. If the human agrees, the robot will grab the object and complete the task. On the other hand, if the human refuses to help the robot, then the robot will ask if the human will place the object. If the human replies with yes, the human will move the object in its final position. Contrarily, the robot says it will try again later.

- **Positioning:** This issue message occurs when a robot needs help precisely positioning an object. The robot will ask the human for help placing the object and thank the human.

After receiving this message, it triggers the node's issue callback function, enabling the callback function to publish a ROS message to the dialogue topic. This initiates the dialogue between the robot and human to allow the human to assist. The dialogue system sends four types of resolution: 1) In the case of human_finish resolution, the human will perform the required work to complete the task. 2) If the resolution is robot_finish, the robot will continue with the remaining work required to finish the task after being briefly assisted by the human. 3) Additionally, if the resolution is collab_finish, the human should simultaneously work with the robot to finish the task. 4) Lastly, the resolution robot_retry happens when the robot must retry the execution and deactivate the node.

This procedure allows the generalized task structure to be utilized in complex, hierarchical tasks that can be prone to failures and require collaboration between humans and robots.

3.4 Interdependence Constraint for Collaborative Multi-Robot Task Allocation

Interdependence constraints were applied to enable explicit communication between multiple robots in the distributed multi-robot task allocation system. An interdependence task is one in which the different parts of the task must be completed simultaneously. Their proposed work focused on tasks like building tasks where one agent holds a part in place while another agent connects another piece.

Because of this reason, the following components were added to the existing multi-robot architecture:

- **WHILE** constraints: This is a newly created goal node that enforces an interdependence constraint on its children. That means one sub task's completion depends on the other sub task's completion.

- **HOLD** behavior: This is a new behavior node that allows one robot to hold an object, while the other robot finishes another part of the task requiring explicit cooperation between the robots. HOLD is designed to be a child of WHILE node only.

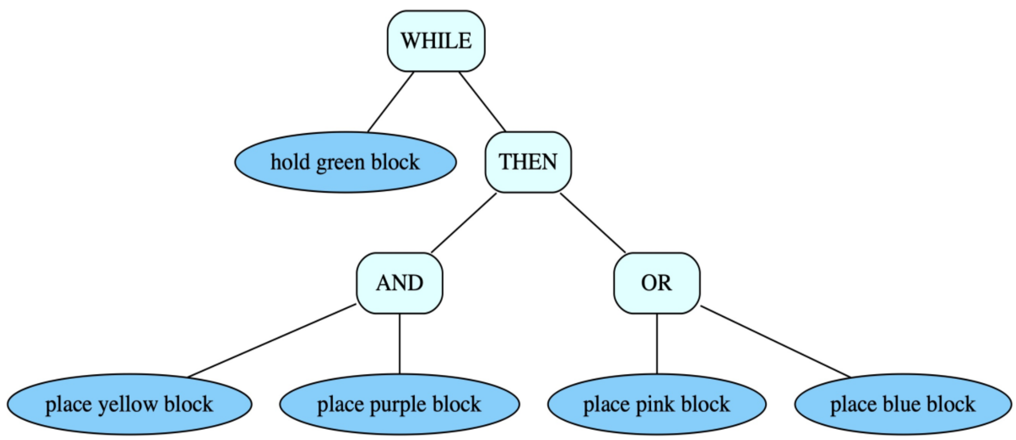


FIGURE 3.3: Interdependence Constraint for Collaborative Multi-Robot Task Allocation

In Figure 3.3, we can see a WHILE functioning task tree with WHILE as a root node. Here, the left child of the WHILE node has the HOLD behavior, and the right child of the WHILE node is the action task that needs to be completed. During this instance, the robot will hold the green block while the action tasks associated with the THEN node along with its sub-tree must be executed.

3.5 Generalized Task Structure Learning

Through human demonstration, a learning framework was integrated into the previous multi-robot architecture, enabling the robots to learn the structure of a complex, hierarchical task. In order to understand a specific task, a human may provide multiple sets of demonstrations of the task. From the demonstration, the first step is segmenting out the specific tasks from the demonstration. Later in the second step, it uses these segmented demonstrations to determine how to execute the task. Lastly, in the third step, the learned task is transferred to the robot to complete the learned task. Previously, they only concentrated on the second step of this framework. The learning framework is developed using a Genetic Algorithm (GA).

3.6 Summary

The previously hierarchical architecture was able to perform tasks consisting of various types of constraints such as sequential (THEN), non-ordering (AND), and alternative paths of execution (OR). Hierarchical task presentations represent the constraints of a task and use the robot's controller to execute it. Each node in the task tree maintains its state information continuously and performs top-down and bottom-up activation spreading to ensure the proper execution of the given tasks with constraints. Later the single robot control architecture was extended to a multi-robot control architecture that supports multiple robots to accomplish a set of tasks.

Here all these robots maintain a copy of the task tree representation identical to the other robots. Therefore, the equivalent nodes across the task trees communicate with each other continuously by message passing.

Furthermore, the multi-robot architecture can have situations where a robot faces a fault while performing a specific task and will require human assistance to complete the task. For this reason, a dialogue-based system was added to the system so that if the continuous fault detection system detects a fault, the robot can communicate with the human to resolve it. Furthermore, new interdependence constraints were also added to the system architecture. This constraint allows one agent to hold an object while the other agent will finish another part of the task.

Moreover, a learning framework was proposed for the architecture to learn a task and integrate the specific task into the system to execute it later. Before, it focused on one step of the framework: using the segmented demonstration to determine how to perform the task.

Chapter 4

Collaborative Human-Robot Hierarchical Task Execution

The fast pace of advancements in the development of autonomous robotic systems opens new possibilities for the use of robots in daily tasks, holding 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 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 chapter, 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.

4.1 Human-Robot Collaborative Architecture

4.1.1 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) [2]. 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. 4.1.

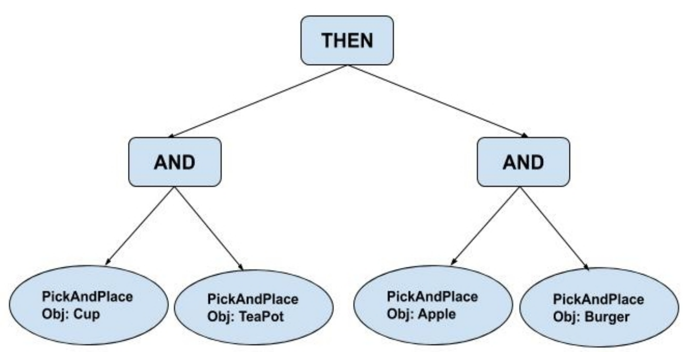


FIGURE 4.1: Hierarchical Task Representation

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 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 [2]

4.1.2 Human-In-The-Loop Hierarchical Architecture

In order to extend the previously developed architecture described in Section 4.1.1 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.4.1. 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. 4.1. 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.

During execution of the task, the robot continuously updates the hand position of the human as shown in Fig. 4.2. 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.

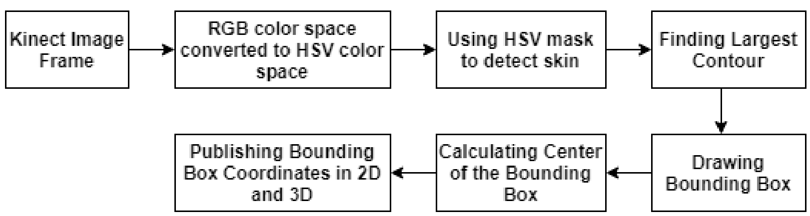


FIGURE 4.2: A step-by-step description of the continuous hand detection system from the Kinect image frame to infer the human intention

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 4.1.

$$SimScore_i = Cosine\_Similarity(\hat{V_{X_i,Y_i,Z_i}}, \hat{V_{x_i,y_i,z_i}}) \tag{4.1}$$

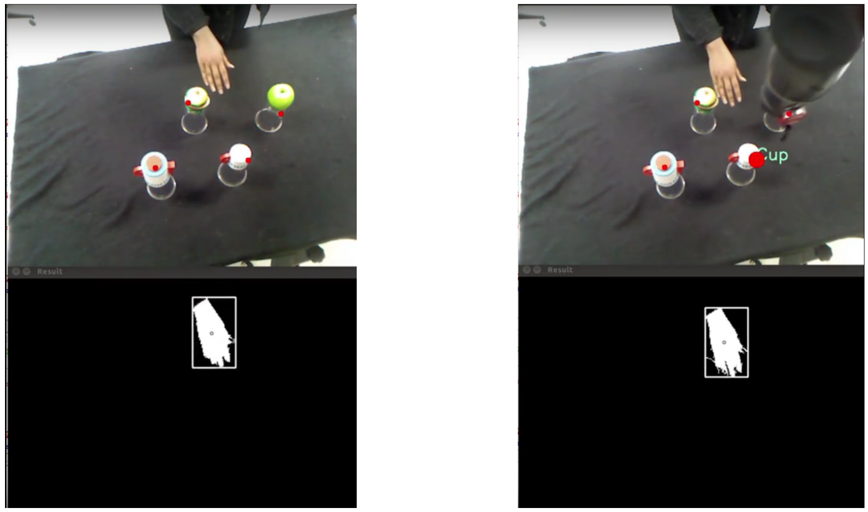
where $\hat{V_{X_i,Y_i,Z_i}}$ and $\hat{V_{x_i,y_i,z_i}}$ are the initial normalized vector and updated normalized vector for object $i \in 1, ..., n$.

- **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 [97].



(A) (B)

FIGURE 4.3: 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.

The messages allow the human architecture to activate an object node when the *Started* value is 1.

4.1.3 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.

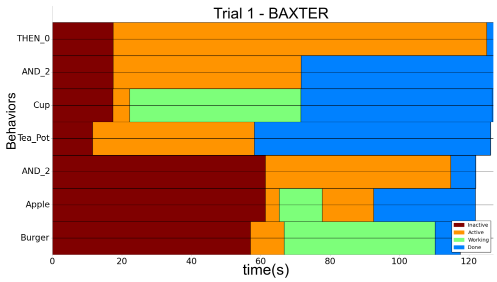
Fig. 4.3 shows the human hand going for the cup during the task. The system hasn't detected the intention yet in Fig. 4.3a. However, in Fig. 4.3b the human's intention can now be inferred and is being shown with a red circle on the object.

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.

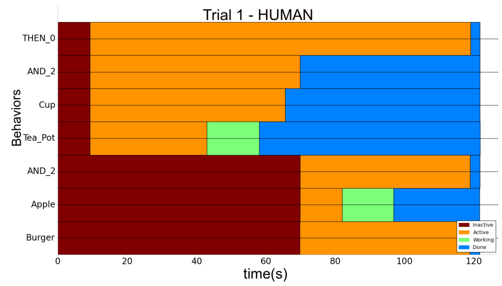
4.2 Experiment Design

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. 4.4. The 3D location of each object is provided by the vision system [98]. 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. 4.1). 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 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.

FIGURE 4.4: 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

FIGURE 4.5: 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 $\rightarrow$ *inactive*, Orange $\rightarrow$ *active*, Green $\rightarrow$ *working*, and Blue $\rightarrow$ *done*.

4.3 Results

The timing diagrams (Fig. 4.5) illustrate the state for each node during scenario execution using the task structures of the human and robot shown in Fig. 4.1. 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. 4.5a). 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. 4.5b, 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. 4.5a).

Fig. 4.5b 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.

4.4 Conclusions and Summary

This chapter 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.

Chapter 5

Homophily and Trust in HRI

People tend to connect with others who are similar to themselves [99]. This tendency, referred by social scientists as homophily, manifests itself with similarities due to gender, national origin, social class background, and other socio-demographic, behavioral and interpersonal characteristics [49]. Individuals in homophilic relationships share common characteristics (such as beliefs, values, education) that make communication and relationship formation easier. In HRI, a robot needs to create a smooth interaction with its audience in order to perform well in social settings. We wish to investigate if robots can benefit from the same social tendency and leverage from homophily in their interactions. We proposed an experiment where a social robot acts in such a way that implies homophily while another robot does not. Then we observed how the person will react toward the robots. We expected that achieving homophily, or bonding based on a common interest or implying similarity, between a human user and a robot, holds a promise of improvement in trust between them.

The similarity between humans and robots is an essential facilitator of positive attitudes toward robots [100]. For instance, Bernier and Scassellati [101] showed that the more an individual believes that a robot is similar to them, the more they like and prefer to interact with them. Also, research of Bowman et al. [100] found that individuals tend to like and build healthier emotional attachment toward robots that appear to have a similar personality to theirs. Finding homophily between individuals is a useful for human-robot interaction. Therefore, we wanted to investigate if this phenomenon could occur between humans and robots as well.

In this chapter, we explore homophily between a person and a robot from a questionnaire by measuring common interest, bonding, and similarity between a person and a robot. The purpose of this work is to determine whether similarities between a robot and a person might improve social connection and trust. If such a link exists, then homophily would be an important physical and behavioral design consideration for effective HRI; this could lead to an improved first impression of a robot, which might eventually help humans communicate and interact with the robot more easily.

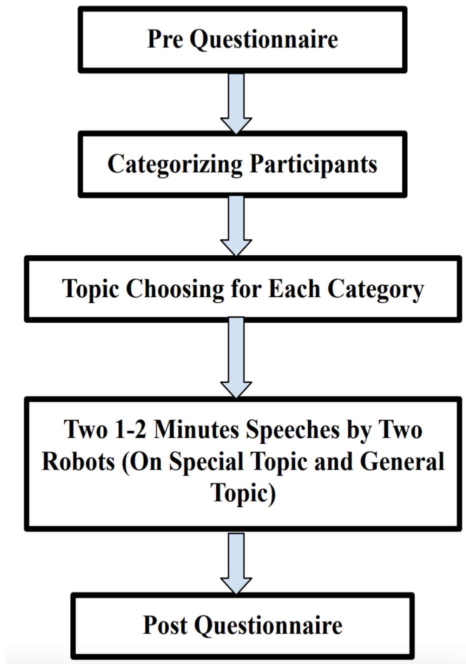


FIGURE 5.1: Proposed Method

5.1 Study Design

In this user study, we aimed to measure the perceived similarities between a person and a robot when they shared a common interest. As our second interest, we were looking into the effect of homophily on trust human-robot trust.

We proposed two hypotheses on similarity and trust:

- **H1**: A person will feel a similarity (homophily) to the robot in a human-robot interaction when they share a common interest
- **H2**: There is a correlation between homophily and trust in human-robot interaction

Our two hypothesis would be tested by making two experimental conditions and analysing data. Our proposed method is divided into a few steps that is shown in Figure 5.1.

5.1.1 Experiment Conditions

In this section, we explain how we developed two conditions for testing out the hypothesis. Each participant experiences condition one in which the person finds similarity to the robot and condition two where it is the opposite. There can be different homophily categorizations based on age, gender, national origin, socioeconomic state, ethnicity, attitude, etc. However, we chose 'National Origin' as our divider for different groups. Since we wanted to find a food known by the person, we considered national origin which means the nation where a person was born, or the country of origin that person's ancestors came from. And, they may know food associated with that area directly or by their family. The correlation between national origin and homophily is also higher than gender [102] for instance. For this study, to more tightly control potential participant differences, we chose only one age range (18-35) and one education level (university students).

The experiment was conducted in a room in one of the libraries on the University of Nevada, Reno campus. For the experiment, we used two NAO robots. We distinguished the robots to the participants as Red NAO and Blue NAO based on their color. Here, the Blue and the Red NAO were the Homophilic Condition Robot and

TABLE 5.1: Homophilic Condition for Each National Origin Category

1.	What is your age?
2	What is your gender? 1.Male 2.Female 3.Other
3.	What is your major and degree?
4.	Are you familiar with robots?
5.	Choose which national origin best represents you: 1. Europe 2. Middle East 3. North African 4. African 5. North American 6. South American 7. Central American 8. Southeast Asia 9. East Asian 10. West Asian 11. Indian 12. Other

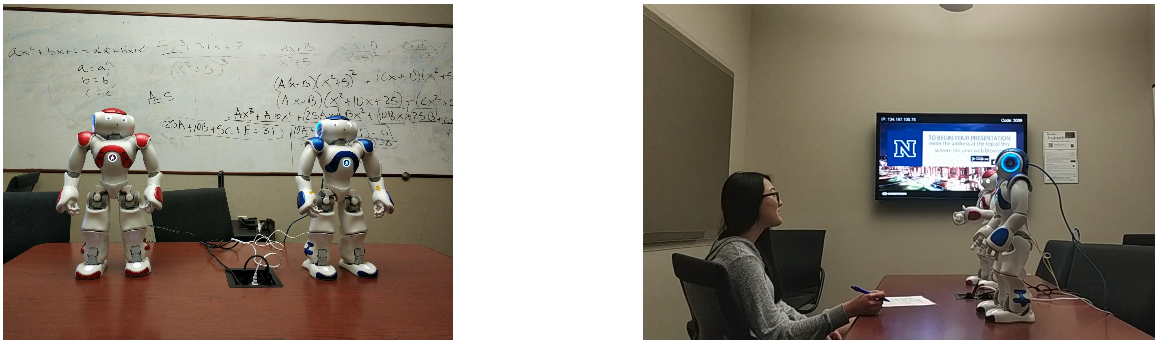
TABLE 5.2: Pre Questionnaire

National Origin	**Homophilic Condition**
Europe	Pirozhki
Middle East	Kebab
North African	Coucous
African	Bobotie
North American	Cheese Steak
South American	Ceviche
Central American	Pupusa
Southeast Asia	Nasi Campur
East Asia	Sichuan Cuisine
West Asia	Kebab
South Asia	Biriyani
Others	Ice Cream

the Non-Homophilic Condition Robot respectively. Fig. 5.2a shows the set up of the robots during the user study. In the pre-questionnaire form (Table 5.2), general information such as age, gender, major, and national origin information were asked of the participant.

5.1.2 Experiment Task

At first, before starting our experiment we explained our experiment in brief to each participant. We let them know that all collected data would remain anonymous. If



(A) Red and Blue NAOs used for the experiment

(B) The participant listening to the robot's speech

FIGURE 5.2: Experimental Setup

the participant agreed to take part in the experiment then we continued with the rest of the experiment.

Our proposed method was divided into 3 major steps. These are: 1) Pre Questionnaire, 2) Speech Presentation, 3) Post Questionnaire

- **Pre Questionnaire:** At first, the participant was given a pre-questionnaire form (Table 5.2) which included demographic questions such as age, gender, major, and national origin information. We used the national origin information to categorize participants.

  We categorized the participants into one of 12 broad national origins: European, Middle East, North African, African, North American, South American, Central American, Southeast Asia, East Asian, West Asian, South Asian, and Other. The name of the national origin category in the U.S. was collected from the United States Census Bureau data [103].

- **Speech Presentation:** We designed an interaction between humans and robots where two NAO humanoid robots gave speech presentations in front of the participant individually (Fig. 5.2b) where the robots were teleoperated by the experimenter from the other room. The participants did not know about the existence of the robot's operator. During each session, one robot gave a presentation on the homophilic condition related to the participant's national origin shown in Table 5.1. After that, the remaining robot gave a presentation on a non-homophilic condition. The topic of the homophilic condition of the presentation for each participant was selected based on the national origin information given by the specific participant in the pre-questionnaire. The famous food dishes from each region of the national origin were chosen as the homophilic condition for each national origin group (Table 5.1). The robot gave a speech presentation on bread as a non-homophilic condition which is familiar to every national origin category.

  Samples of the speeches by the homophilic condition robot and the non-homophilic condition robot are given below respectively, where the homophilic condition robot's speech is about 'Kebab' towards the participants categorized into the 'Middle East' and the non-homophilic condition robot's speech is about 'Bread.'

  - Homophilic Condition Robot: *'Hi, I am Blue NAO. I am going to talk about a dish named Kebab. Kebab is a very popular dish all around the world. Shish Kebab or doner Kebab can be two familiar names of Kebab. It is often served during special occasions. It can be made with ground meat* *or seafood, even sometimes with fruits and vegetables. Traditional meat of Kebab is most often mutton or lamb, but regional recipes may include beef. Sometimes Onions are often added with Kebab to enhance the taste. Kebab is served with various dishes according to each recipe. Kebab with naan is very popular in some regions. Yogurt drink is often served with Kebab. It is also served with rice, grilled tomatoes, tabbouleh salad, or bread. There are many restaurants in Reno where we can find Kebab, and they are delicious. Well, I hope you enjoyed my speech.'*

– Non-Homophilic Condition Robot: *'Bread is a staple food prepared from a dough of flour and water, usually by baking. Throughout recorded history, it has been popular around the world and is one of the oldest artificial foods, having been of importance since the dawn of agriculture. Proportions of types of flour and other ingredients vary widely, as do modes of preparation. As a result, types, shapes, sizes, and textures of bread differ around the world. Bread may be leavened by processes such as reliance on naturally occurring sourdough microbes, chemicals, industrially produced yeast, or high-pressure aeration. Some bread is cooked before it can leaven, including for traditional or religious reasons. Non-cereal ingredients such as fruits, nuts and fats may be included. Commercial bread commonly contains additives to improve flavor, texture, color, shelf life, nutrition, and ease of manufacturing. Also, bread has a social and emotional significance beyond its importance as nourishment. It plays an essential role in* *religious rituals and secular culture. Well, I hope you enjoyed my speech.'*

- **Post Questionnaire:** Each speech took less than 3 minutes. After listening to these presentations one after another, the participant filled out a post-questionnaire form. There were questions regarding homophily, trust, and provided speeches. The questionnaire was divided into two parts. First part was observing the effect of the speech on the trust by asking each participant to choose one of the robots to pick one snack for themselves from the other room.

  The other part consisted of questions to measure the degree of both homophily and trust (see Table 5.3). This questionnaire was adapted from [104] and Jian et al.[105] to measure homophily and trust respectively. We also added some extra questions related to this experiment that would help us to analyze the answers. All the questions in the questionnaire are based on five-point Likert scale.

5.2 Results and Analysis

Details of experiment results and analysis are presented in this section. We analyzed data from questionnaires in order to support or refute our hypotheses presented above.

Participants were gathered from the University of Nevada, Reno campus area. Most of the participants' age ranged from 18 to 35. We initially recruited 19 participants, and discard three participants' data due to robot malfunctions. We used the remaining

Table 5.3: Post-Questionnaire

Category	Question	Type
Homophily	The Robot was similar to me	(1-5)
	The Robot thinks like me	(1-5)
	The Robot behaves like me	(1-5)
	The Robot and I had a common interest	(1-5)
	I felt a bond with the Robot while it was speaking	(1-5)
Being Suspicious	The Robot is deceptive	(1-5)
	The Robot behaves in the underhanded manner	(1-5)
	I am suspicious of the Robot's intent,action or outputs	(1-5)
	I am wary of the Robot	(1-5)
	The Robot's actions will have a harmful or injurious outcomes	(1-5)
Security	I am confident in the Robot	(1-5)
	The Robot provides security	(1-5)
Trust	The Robot is dependable	(1-5)
	The Robot is reliable	(1-5)
	I can trust the Robot	(1-5)
Familiarity	I am familiar with the Robot	(1-5)
Topic	Are you familiar with the blue Robot talked about?	(1-5)
	Which speech did you find more interesting?	(1-5)

Table 5.4: One-Sample Test (Test Value = 3)

	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval	
					Lower	Upper
Common Interest	4.858	15	0.000	0.938	0.53	1.35
Felt Bonding	2.551	15	0.022	0.688	0.11	1.26
Similarity	3.162	15	0.006	0.500	0.16	0.84

16 participants in our analysis, 6 male, and 10 female. Among the participants, there were 4 participants from Southeast Asia, 4 participants from the Middle East, 3 participants from South Asia, 2 participants from East Asia, 2 participants from

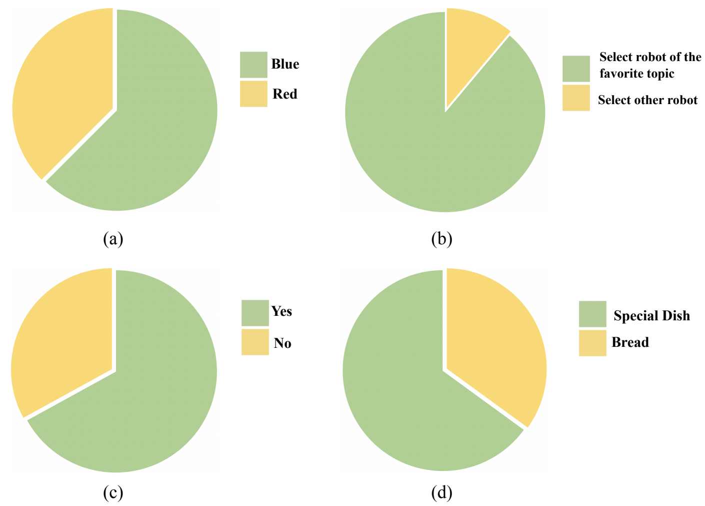


FIGURE 5.3: (a) Chosen Robot, (b) Familiarity with topics, (c) Chosen robot is the one with dish topic, (d) More interesting topic

North America, and 1 participant from Europe.

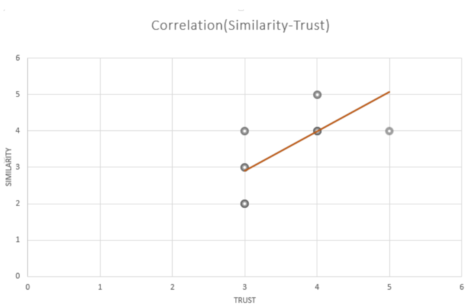
We explored results related to our hypothesis: first, homophily among participants (two groups of the ones who chose the Blue NAO and those who chose the Red NAO); second, the correlation between homophily and trust categories in data.

To have a better understanding of our data, we used pie charts. The data shown in figures 5.3(a)-(d) relate to our experiment hypotheses. The majority of the participants (62.5%) chose the blue robot (homophily condition) in the first part of the post-questionnaire which we mentioned in Section 5.1.2.

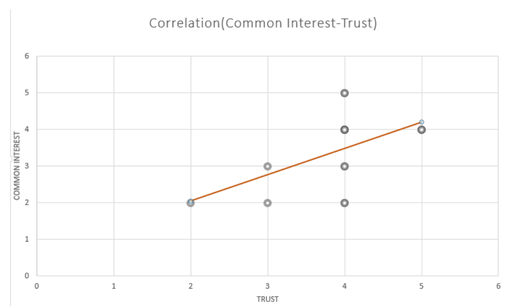
We further investigated why some participants preferred the red NAO. Many countries share one origin, but there is a possibility that people of one origin may not be familiar with exceptional food. For those participants with no idea about the unique food, the general topic of 'bread' the familiar topic. Fortunately, The last two questions in the 'topic' category of post-questionnaire shown in TABLE 5.3 define this issue and clear

TABLE 5.5: Correlation

		Reliability	**Trust**	**Similarity**	**Common Interest**
Reliability	Pearson Correlation	1	.631**	0/316	-0/022
	Sig. (2-tailed)		0/009	0/233	0/937
	N	16	16	16	16
Trust	Pearson Correlation	.631**	1	.665**	.539*
	Sig. (2-tailed)	0/009		0/005	0/031
	N	16	16	16	16
Similarity	Pearson Correlation	0/316	.665**	1	0/205
	Sig. (2-tailed)	0/233	0/005		0/447
	N	16	16	16	16
Common Interest	Pearson Correlation	-0/022	.539*	0/205	1
	Sig. (2-tailed)	0/937	0/031	0/447	
	N	16	16	16	16



(A) Similarity-Trust



(B) Common Interest-Trust

FIGURE 5.4: Correlation

if the person is familiar with the blue NAO topic or not, and which topic was more interesting for him/her. So, we used the favorite topic question to compare 'chosen robot' and 'favorite topic' to have a new query, which is 'the participants whose choice was in line with their favorite topic. If choosing (Red NAO-homophily condition) and (Blue NAO-homophily condition), the person gets a 1 and otherwise gets a 0. We observed this group owned 80% of the population (see Figure 5.3(c)). We conclude that participants mostly chose the robot that was talking about a familiar topic.

To investigate our first hypothesis for each independent variable, we analyzed the results using one sample t-test, knowing that the experiment has one sample group with two variables. As seen in Table 5.4 a one-sample t-test showed that there is a significant difference in mean 'common interest' between the homophilic and non homophilic conditions ($p < .001$). There was a significant difference in mean 'felt bonding' between the the homophilic and non homophilic conditions ($p < .001$). There was also a significant difference in mean 'similarity' between the homophilic and non homophilic conditions ($p < .001$) (see Table 5.4).

To explore our second hypothesis, we used Pearson correlation test results (see Table 5.5). We found that there is a moderate positive correlation between 'similarity' and 'trust' variables ($r = 0.665, n = 16, p = 0.005$) (see Figure 5.4a). There was also a moderate positive correlation between 'Common Interest' and 'Trust' ($r = 0.539, n = 16, p = 0.03$) (see Figure 5.4b).

5.3 Conclusion and Summary

In this chapter, we explored the effect of national origin as homophilic condition in case of Human-Robot interaction because among all of these 'national origin' is a significant social divider today [106].

Our two hypotheses were supported by our results shown in the prior section. Our first hypothesis, H1: **"A person will feel a similarity (homophily) to the robot in a**

human-robot interaction when they share a common interest" was supported via the significant result in the similarity comparison shown in Table 5.4. H2: **"There is a correlation between homophily and trust in human-robot interaction"** was supported by showing that there is a correlation between homophily and trust in human-robot interaction in Table 5.5. The responses to question one show the preference for the homophily condition with a correlation for preference in the robot with familiar topic (see Figure 5.3). This question gave participants a forced choice between robots to pick their prize (snack), which reflects trust in a social situation. We also asked our participants to explain their reasoning after choosing a robot, and most of the comments showed that they were trusting the robot that shares the interest or the topic robot was talking about was more familiar to them. This 'trust' can be contextualized with two comments: "If he were talking about bombs, I would have not to trust him, but he was talking about Biryani! I love spicy food."; "I chose the blue one because I love kebab, and I miss it."

There is room for more investigation on our proposed hypotheses by having more participants. We can have more accurate homophily categories and related speech for each category. That will profoundly affect our results because the more robot's speech is close to a person's homophily group; our results can reflect the more accurate result.

Chapter 6

Learning a Hierarchical Task Structure for Human-Robot Teams

An effective human-robot collaborative task architecture is efficient, flexible, and dynamic. For robots to perform tasks concurrently with human partners, some special abilities or features are expected from the robots. For example, humans and robots will have to interact to accomplish complicated tasks that are typically done with human-human teamwork. While it may be easy for one human teammate to teach another human teammate about the task or to provide input to a revised task design, the situation in the human-robot domain is different. Typically, to learn or update a task, the robots need to be programmed by a robotics expert or programmer manually. If the robot could learn directly from a teammate, training the robot to perform new tasks would be more efficient and functional.

Learning a new task from human demonstration consists of several significant components. To learn a particular task, a human may provide a robot with several demonstrations of the task. Given these demonstrations, individual tasks must be segmented out. For this paper, the individual tasks are a particular object's pick and place movements. Next, the relationships between segmented demonstrations are used to learn how to perform the task. In our case, this entails learning the sequence in which the objects were placed. The last step is transferring these learned tasks to the robot to ensure the robot correctly learned the task. In this work, the robot uses the demonstrations to construct a hierarchical task tree and can directly execute the learned tasks.

A single demonstration is represented by a particular ordering in which objects are placed. Due to the constraints inherent to a generic task, there may be multiple ways to perform a given task, such as in a building task. Therefore, the learning scheme must encompass the set of possible orderings within a single task structure. The proposed work represents this set of sequences with a hierarchical task representation consisting of a set of constraints (ordering, non-ordering, and multiple paths of execution). Several assumptions regarding the demonstrations are made. The demonstrations provided must represent a fully completed task. To learn the ordering constraint, we also incorporate a set of bad demonstrations representing incorrect ways to perform the task. These bad demonstrations can be either completed tasks or partial tasks.

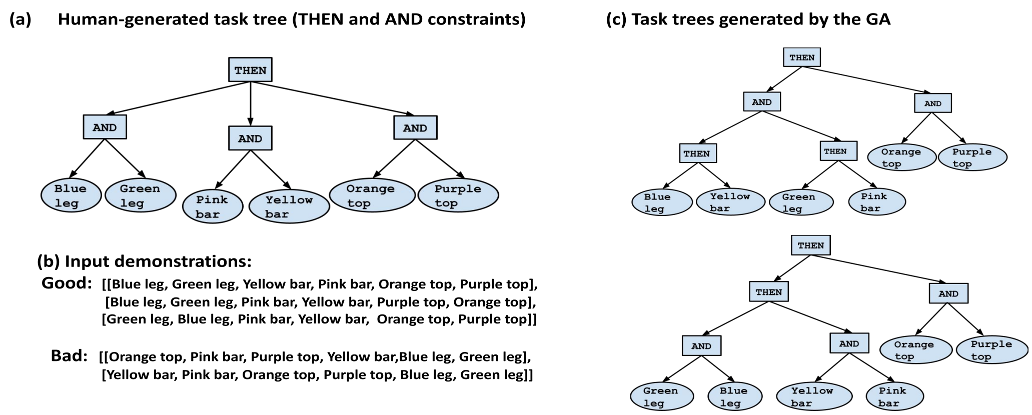


FIGURE 6.1: (a) Human-generated task tree to assemble the tabletop with six objects using *THEN* and *AND* constraints. (b) Three good and two bad demonstrations were provided to the GA. (c) Two example hierarchical representations generated by the GA from the given input demonstrations. Both trees reflect the constraints inherent to the human-generated task tree.

This paper proposes a pipeline for teaching robots tasks through a human task interface. This interface allows the robot to monitor the human teammate's movement and converse with them in natural language. The robot is able to learn the task demonstrated through a genetic algorithm-based framework. This framework generates a hierarchical task structure representation, which accurately represents the constraints inherent in the demonstrations. This approach reduces the need for a robotics expert or programmer to manually program new tasks and instead allows non-programmers to teach new tasks to the robots.

6.1 Task Structure Learning Framework

The goal of the task structure learning framework is to learn a hierarchical representation that accurately represents the constraints inherent in a task. This framework utilizes our previous hierarchical task architecture [107] as the basis for the hierarchical representations that represent the tasks and their inherent constraints. Therefore, the relevant details of our previous work are discussed briefly in Section 6.1.1.

6.1.1 Hierarchical Task Representation

Real-world tasks can be a collection of sequential, non-sequential, and alternative subtasks. Our robot control architecture currently enables the system to encode tasks implicating various types of constraints such as sequential (THEN), non-ordering (AND), and alternative paths of execution (OR)[107]. The task structure is a tree that has the task components to be completed as leaf nodes and the constraints between these components as internal nodes. The complete details of the hierarchical architecture are described in [107], extended to work for human-robot teams by understanding human intention [96] as well as utilizing dialogue to allow a human to assist the robot during fault recovery [95].

6.1.2 Genetic Algorithm Framework

The proposed framework is based on a genetic algorithm (GA), discussed in Sections 6.1.2.1-6.1.2.3. The GA is used to learn a hierarchical representation from a set of demonstrations. These demonstrations represent various orderings in which the objects can be placed for a particular task, as seen in the building experiment task (Fig. 6.1).

6.1.2.1 Compression-based Encoding Scheme

The purpose of the GA is to generate chromosomes that represent a hierarchical task structure representation for a given task. This work assumes that the representations are binary trees. The goal is to generate simple, compact, human-interpretable trees in which each object for a given task appears only once. Given these two restrictions, a generated task tree can have at most (*n-1*) constraints (internal nodes) where n is the number of objects in the task. To handle the complexity of large tasks, a compression-based encoding scheme was designed to maintain a consistent size of chromosomes.

The compression-based encoding uses a dictionary to store the compressed chromosomes. To simplify the encoding, each of the n objects in a task is mapped to unique numbers from *1* to n to allow demonstrations to be represented by a numerical sequence instead of a set of words. Initially, the dictionary only contains the set of objects *1* to n for a given task.

The GA generates chromosomes of the form (*number_left, constraint, number_right*) where *number_left* and *number_right* are two different numbers in the dictionary and *constraint* is one of the constraints handled by the hierarchical representation (*THEN, AND, OR*). In the initial population of the GA, the chromosomes are built entirely from objects and a single constraint.

At each generation, new chromosomes with high fitness (Section 6.1.2.2) will be added to the dictionary under a unique numeric representation (*n+1*, etc.).

Therefore each of the *number_left* and *number_right* in later generations can represent encoded chromosomes previously stored in the dictionary.

Using this compression-based encoding, the chromosomes generated by the GA will always be of the form *(number_left, constraint, number_right)* while continuing to produce more complex encodings with each generation. After the GA finishes, the dictionary can be used to decode the compression in order to get the complete hierarchical representation consisting solely of the base numbers (*1-n*, corresponding to the objects) along with the constraints generated between them.

6.1.2.2 Fitness Function

The generated hierarchies are scored via a fitness function that takes into account the desired structure of the hierarchical task representation. These trees should be compact and represent the complete set of constraints inherent to a task. The correctness of the trees is determined by evaluating how well the tree reflects the provided demonstrations and their constraints.

By looking solely at good demonstrations, it is very difficult to differentiate between the *THEN* and *AND* constraints. Therefore, a set of bad demonstrations, providing incorrect orderings of the task, is also included to provide a notion of ordering between objects in the task to allow the GA to learn the *THEN* constraint easier.

The fitness function used in the GA is defined in Equation 6.1.

$$\begin{aligned} score = & \; count_{good} * multiplier \\ & - count_{bad} * multiplier \end{aligned} \tag{6.1}$$

$$\begin{aligned} multiplier = & (multiplier_{left} \\ & + multiplier_{right}) * w_{constraint} \end{aligned} \tag{6.2}$$

$$where\ w_{constraint} = \begin{cases} 4 & THEN \\ 1 & AND \\ 1 & OR \end{cases}$$

The fitness function evaluates how well a chromosome fits the demonstrations weighted by a multiplier which accounts for the combinations of constraints within the chromosome. At each depth in the tree, the multiplier is calculated as in Equation 6.2. Each type of constraint has a different multiplier value associated with it. The $multiplier_{left}$ is the multiplier for the left sub-tree at a given level and $multiplier_{right}$ is for the right sub-tree. The *THEN* constraint has a higher multiplier than the *AND* and *OR* nodes to allow the GA to better learn the *THEN* constraints, which are difficult to illustrate through demonstration alone.

The fitness function also accounts for the number of demonstrations whose constraints are accurately reflected by the chromosome. The $count_{good}$ represents the number of good demonstrations that fit the chromosome's constraint and the $count_{bad}$ represents the number of bad ones which do not. The fitness function (Equation 6.1) uses the weighted difference between these two values to ensure that the chromosome fits the good demonstrations but not the bad ones. If the fitness score is above a certain threshold (i.e. number of good demonstrations), the chromosome gets saved as a new rule in the dictionary. This rule can then be used to generate new chromosomes in future iterations.

How well a chromosome fits a demonstration is determined by evaluating if the ordering and constraints between the objects in the demonstration are accurately reflected in the chromosome. This is done by checking whether the *number_left* and *number_right* occur in the correct order in a given demonstration for a certain constraint (*THEN, AND, OR*). For the *THEN* constraint, *number_left* must come before *number_right* in the demonstration. For the *AND* constraint, the order of the numbers doesn't matter, but both *number_left* and *number_right* must appear in the demonstration. For the *OR* constraint, either *number_left* or *number_right* can appear in the demonstration, but not both. If these cases are not met, the corresponding constraint function returns a fail, meaning that it does not fit the demonstration's constraints.

These constraints are verified by recursing through the compressed chromosomes to evaluate all of their constraints. If a chromosome contains more than one instance of each object, it also returns a fail. This ensures that not only are the constraints met but that the representations are as compact as possible, with a tree depth of at most *n-1* due to their binary nature.

Algorithm 1: Modified genetic algorithm for learning a hierarchical task structure representation

```
Generate initial population of k individuals.
for each individual in the population do
    Evaluate the fitness
    if fitness > THRESH then
        Add individual to dictionary
    else
        Add individual to bad dictionary
    end if
end for
while i < MAX_ITERS do
    Select top 70% of previous population as offspring
    Generate other 30% of offspring via the following:
    Generate 15% new individuals with any numbers in the dictionary as number_left and number_right
    Generate 15% new individuals with simple numbers corresponding only to the objects as number_left and
    number_right
    Cross over offspring to form new offspring with probability p_c
    Mutate new offspring with probability p_m
    Place new offspring in a new population
    for each individual in the population do
        Evaluate the fitness
        if fitness > THRESH then
            Add individual to dictionary
        else
            Add individual to bad dictionary
        end if
    end for
end while
Find the individual with highest fitness in dictionary
Decode individual to get the complete hierarchical plan
```

6.1.2.3 Modified Genetic Algorithm

The framework uses a modification of the standard GA method consisting of two major changes: 1) each generation, the chromosomes which have a fitness above a certain threshold are added to the dictionary which contains the compressed chromosomes (Sections 6.1.2.1 and 6.1.2.2) and 2) the best resulting chromosome at the end of the GA must be decoded using the dictionary to get the complete hierarchical plan ((Sections 6.1.2.1). The modified algorithm is presented in Algorithm 1. The main implementation for the GA was done using the DEAP framework [108]. Individuals were generated using *(number_left, constraint, number_right)* as described in Section 6.1.2.1. Evaluation of the fitness was done as described in Section 6.1.2.2. Selection, Algorithm 1 line 11, was performed through the roulette selection method. The crossover in Algorithm 1 line 15 was performed with the one-point crossover method.

The mutation in Algorithm 1 line 16 was performed using the method described in Algorithm 2. The algorithm mutates one of the components of an individual: *number_left*, *number_right*, or *constraint*. Randomized positions 1 to 4 allow the mutation of the *number_left* and *number_right* 25% of the time and the *constraint* the other 50%.

6.2 GA Validation

The GA framework for generating hierarchical representations from a set of demonstrations (Section 6.1.2) is validated on a building task. A human-generated hierarchical representation was created for this task to act as ground truth. The GA

Algorithm 2: Mutate (individual)

```
mutated = individual
Generate random position pos for mutation from 1 to 4
if pos == 0 then
    generate random number mut from 0 to length of dictionary
    while mut == individual.number_left or mut == individual.number_right do
        generate a new random number mut from 0 to length of dictionary
    end while
    mutated.number_left == mut
else if pos == 2 then
    generate random number mut from 0 to length of dictionary
    while mut == individual.number_left or mut == individual.number_right do
        generate a new random number mut from 0 to length of dictionary
    end while
    mutated.number_right == mut
else if pos == 1 or pos == 3 then
    generate constraint type const out of {THEN, AND, OR} with probabilities (66%, 33% ,33%) respectively
    mutated.constraint == const
end if
RETURN mutated
```

framework was run ten times on this task to evaluate whether the GA is able to successfully and consistently generate a hierarchical representation that meets the constraints defined in the human-generated tree. The details of the experiment are shown in Fig. 6.2. The task includes a combination of the three constraints (*THEN*, *AND*, *OR*) as shown in Fig. 6.2(a). The task tree was used to generate both a set of good demonstrations which represent valid orderings for each tree and a set of bad demonstrations which represent invalid orderings, or orderings that break the constraints of the tree as shown in Fig. 6.2(b). These bad demonstrations can be either full sequences (containing a full ordering of objects) or partial sequences (containing a partial ordering of a few objects). These demonstrations are passed as input to the GA in order to learn a hierarchical representation that reflects the inherent task constraints. The human-generated task tree is not given to the GA as input. Instead, it is used to verify whether or not the task trees generated by the GA reflect the same constraints as those provided by humans. This verification is done by hand. The top

(a) Human-generated task tree

((((PLACE greenleg) AND (PLACE blueleg)) THEN ((PLACE yellowbar) AND (PLACE pinkbar))) THEN ((PLACE orangetop) OR (PLACE purpletop)))

(b) Input demonstrations:

Good	[Blue leg, Green leg, Yellow bar, Pink bar, Orange top]
	[Blue leg, Green leg, Pink bar, Yellow bar, Purple top]
	[Green leg, Blue leg, Pink bar, Yellow bar, Orange top]
Bad	[Pink bar, Purple top, Yellow bar,Blue leg, Green leg]
	[Yellow bar, Pink bar, Orange top, Blue leg, Green leg]

(c) Validating task trees generated by the GA

(((((PLACE green_leg)AND(PLACE blue_leg))THEN(AND(PLACE yellow_bar)(PLACE pink_bar)))THEN(OR(PLACE orange_top)(PLACE purple_top)))	Correct
(((PLACE blue_leg)AND(PLACE green_leg))THEN(((PLACE yellow_bar)AND(PLACE pink_bar))THEN((PLACE orange_top)OR(PLACE purple_top))))	Correct
((((PLACE blue_leg)AND(PLACE green_leg))THEN((PLACE pink_bar)AND(PLACE yellow_bar)))THEN((PLACE orange_top)OR(PLACE purple_top)))	Correct
((((PLACE blue_leg)AND(PLACE green_leg))THEN((PLACE yellow_bar)AND(PLACE pink_bar)))THEN((PLACE orange_top)OR(PLACE purple_top)))	Correct
((((PLACE blue_leg)THEN(PLACE pink_bar))AND((PLACE green_leg)THEN(PLACE yellow_bar)))THEN((PLACE purple_top)OR(PLACE orange_top)))	Correct
((((PLACE blue_leg)AND(PLACE green_leg))THEN((PLACE yellow_bar)AND(PLACE pink_bar)))THEN((PLACE purple_top)OR(PLACE orange_top)))	Correct
(((PLACE green_leg)AND(PLACE blue_leg))THEN(((PLACE pink_bar)AND(PLACE yellow_bar))THEN((PLACE purple_top)OR(PLACE orange_top))))	Correct
(((PLACE blue_leg)AND(PLACE green_leg))THEN(((PLACE yellow_bar)AND(PLACE pink_bar))THEN((PLACE orange_top)OR(PLACE purple_top))))	Correct
((((PLACE blue_leg)THEN(PLACE yellow_bar))AND((PLACE green_leg)THEN(PLACE pink_bar)))THEN((PLACE purple_top)OR(PLACE orange_top)))	Correct
(((PLACE green_leg)AND(PLACE blue_leg))THEN((PLACE purple_top)OR(((PLACE yellow_bar)AND(PLACE pink_bar))THEN(PLACE orange_top))))	Incorrect

FIGURE 6.2: Example GA experiment for validation of GA generated task tree with all three constrain types (*THEN*, *AND*, *OR*). (a) The human-generated task tree was used as the ground truth for the experiment. (b) The demonstrations (both good demonstrations and bad demonstrations) were used as input to teach the GA. (c) 10 sample task trees generated by the GA were used to validate the GA. It was observed that 9 out of 10 samples were correct which means they were reflecting the same approach as the human-generated task tree. Due to the random chance of the GA, there is a possibility that a small number of input demonstrations might not be sufficient to capture the desired constraints of a task, as seen in the last validation tree.

task tree generated by the GA on the ten runs is shown in Fig. 6.2(c). Nine out of ten runs resulted in a correct hierarchical representation. However, due to the random chance of a GA, it is possible that a small number of human demonstrations and/or running the GA for a small number of generations may not sufficiently encompass the knowledge required to learn the entire set of constraints of a task. This is reflected by the last validation run shown in (c). However, this validation shows that the GA is able to learn hierarchical tasks with complex constraints using a set of demonstrations relatively accurately and consistently. The GA method is further applied in the human-robot task illustrating our proposed pipeline in Section 6.1.2.3.

6.3 Task Training Pipeline and Experiment

To showcase the capabilities of our proposed system, we have divided our framework into three parts: task demonstration, task learning, and task execution.

6.3.1 Human Task Tracking and Task Encoding

In this module, the robot will track the human teammate's movements while performing a task and record the information as task sequences. The robot will monitor a set of good task demonstrations and a set of bad task demonstrations that serve as input for the next module.

This module was demonstrated in a lab environment experiment with a human partner and a Baxter humanoid robot standing in front of a table with objects. The task for our experiment was to assemble a tabletop with six different parts, each with a different color. A Kinect v2 camera was mounted on Baxter's head to monitor the human teacher's movements using color detection. By tracking each color, the system knows when one object is moved and used to assemble the tabletop. This way the robot can record the order in which the objects are moved into a task sequence.

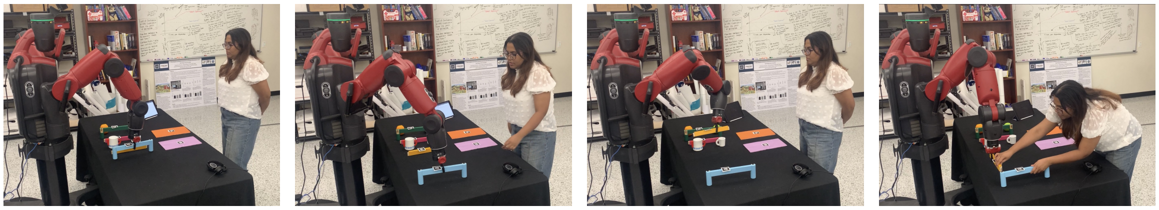
Before tracking and recording each task sequence, the robot asks the human teacher if they are ready to perform a good task demonstration. Upon receiving confirmation from the human user, the robot starts monitoring and saving the task sequences based on the object's movements. When the robot has taken a set of good task sequences, it will again ask the individual if they are willing to demonstrate a bad task. After human confirmation, the robot will track and record the bad task sequences in the same way as before.

6.3.2 Using the GA Framework to Learn a Task

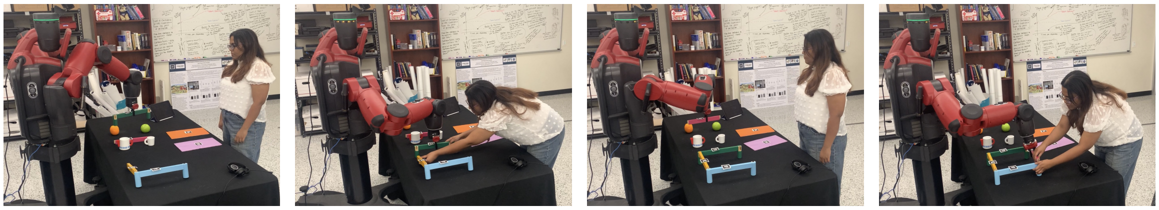
In this module, we apply the GA (Section 6.1.2 to the collected task demonstrations from the previous step. After applying the GA method, the robot will ask the human user if the resulting hierarchical task structure representation is correct or not. The resulting task structure is shown on a web page that the human can see on a nearby tablet. We performed this task by publishing the result string using roslibj, a standard ROS JavaScript library. The framework will move to the next step if a human user confirms the resulting task structure to be correct by saying *"Yes."*. Otherwise, the robot will repeat the GA process.

6.3.3 Executing the Generated Task Structure

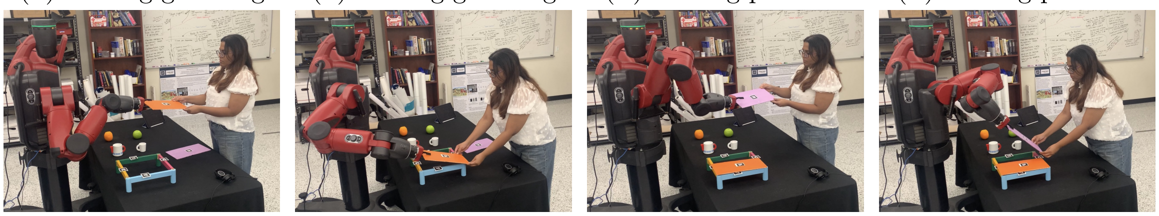
Once the hierarchical task structure is generated, the robot can interpret this structure as a controller as described in [107] and use assemble the tabletop with the human user's assistance. Different AR tags [109] are used to identify the objects and determine their location by tracking their pose. A Kinect v2 camera on top of Baxter's head and Baxter's right-hand camera were used to detect the object's AR tags.



(A) Picking blue leg (B) Placing blue leg (C) Picking yellow bar (D) Placing yellow bar



(E) Picking green leg (F) Placing green leg (G) Picking pink bar (H) Placing pink bar



(I) Asking for the orange top (J) Placing the orange top (K) Asking for the purple top (L) Placing the purple top

FIGURE 6.3: Step by step task execution according to the GA generated task tree. The robot is assembling the table top using the object on the table with the help of the human user

6.4 Results and Discussion

The results of the validation experiment of the GA method are discussed in detail at the end of Section 6.2. For the task training pipeline experiment (Fig. 6.1 and Fig. 6.3), the robot successfully collected three good task sequences and two bad task sequences to assemble the tabletop. Each time the robot begins recording a task demonstration, it asks *"Are you ready to demonstrate how to assemble this tabletop? Please answer "Yes" or "No".* When the human teacher confirms with *"Yes"*, the robot will start tracking the teacher's movement and recording the task sequence. This process was validated by placing each of the tabletop objects at a different location each time. Objects not part of the tabletop were also located on the table to show the robustness of the object tracking.

The collected sequences were provided to the GA to learn a hierarchical task structure representing the task. As in the validation done for the GA in Section 6.2, the task structure and demonstrations for this experiment were also run through the GA multiple times for validation. Two resulting task structures are shown in Fig. 6.1(c).

The robot-generated task tree chosen for this experiment was: ***((((PLACE blue leg) THEN (PLACE yellow bar)) AND ((PLACE green leg) THEN (PLACE pink bar))) THEN ((PLACE orange top) AND (PLACE purple top)))***.

Once the robot finished learning the task structure, it asked the human to verify the display of the GA-generated task tree on a nearby tablet.

After verifying the task tree, the human user confirmed it by saying *"Yes"* to the robot. Following that, the robot responded with *"Thank you. Please help me to assemble the tabletop"* in order to ask the human for assistance in executing the learned task. The task execution is shown in Fig. 6.3. The robot first picks and places the blue leg according to the task tree (Fig. 6.3a and Fig. 6.3b respectively). Next, it picks the yellow bar (Fig. 6.3c and connects it with the blue leg with the help of the human (Fig. 6.3d). After adding the yellow bar, the robot goes for the green leg (Fig. 6.3e) and places it on the yellow bar with further assistance from the human (Fig. 6.3f). The robot completes the table base by picking and placing the pink bar with human help (Fig. 6.3g and 6.3h respectively). Afterward, the robot began picking the orange top. However, since the orange top was out of reach from the robot, the robot asks the human, *"Can you please bring me the orange top? It is out of my reach."* (Fig. 6.3i). After placing the orange top (Fig 6.3j), the robot asks the human for help with the purple top, *"Can you please bring me the purple top? It is out of my reach."* (Fig. 6.3k). Finally, it takes the purple top from the human and places to complete the tabletop task (Fig. 6.3l).

This experiment demonstrates the proposed pipeline for teaching robots new tasks through demonstration, verbal cues, and a GA framework that represents the task as a hierarchical task structure. This process is done without any modification from a programmer, which shows that it can be more flexible, efficient, and dynamic for allowing non-programmers to teach robots new tasks.

6.5 Summary

This chapter proposes a pipeline for teaching robots tasks through a human task demonstration interface that relies on social cues and a GA-based learning framework that emphasizes learning a hierarchical task structure representation. Using this pipeline, the human is able to teach a robot to perform tasks with complex, hierarchical constraints.

The robot tracks and records the human teacher's task sequences using a color detection method in the human task demonstration interface. Additionally, the robot holds conversations with the human to notify them when it begins tracking the sequences and to ask if the human is ready or not. All the sequences serve as inputs for the genetic algorithm-based learning framework. The learning framework generates the structure of a complex, hierarchical task through a small number of human demonstrations. The proposed method uses a hierarchical task representation (Section 6.1) which is able to represent the various constraints of a task such as sequential, non-ordering, and multiple paths of execution. The GA framework is able to learn the constraints of the task in a compact task structure. This task structure can then be executed directly by the robot.

The proposed work is validated in two experiments. The first experiment validates the GA-based method on a building task and was shown to produce a representation similar to the tree generated by the human nine times out of ten. The second experiment validates the proposed pipeline. In this experiment, a hierarchical task structure representing a building task was created by a human. The human provided three good and two bad demonstrations to the robot using the human task demonstration interface. These demonstrations were then used by the robot to run the GA-based learning framework which was able to generate a hierarchical task structure. The robot successfully executed the learned task.

Our experiments show that the robot can learn all the constraints inherent to demonstrations provided by a human in a complex, hierarchical task. These demonstrations were provided through verbal and visual cues which illustrate how the proposed work could allow non-programmers to handle teaching robots new tasks.

Chapter 7

Cognitive Approach to Hierarchical Task Selection For Human-Robot Interaction in Dynamic Environments

Recent advancements in the development of intelligent robots have opened opportunities in which humans and robots can work collaboratively in dynamic environments [1, 110]. Consequently, it increases the need for robots and other agents to understand implicit and explicit cues from teammates and convert these cues into appropriate actions [1, 96]. The importance of understanding the environment may be justified by the following examples from Human-Human-Interaction (HHI): if we can say to our

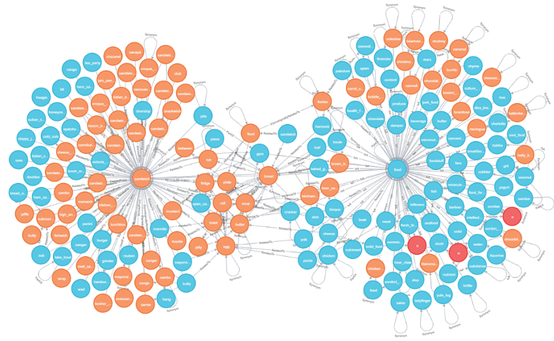


FIGURE 7.1: Sub Graph from Semantic Memory: Semantically populated Networks of "Sandwich" and "Food" nodes

teammate (human) that "It is getting hot outside" or "I am feeling thirsty" rather than "I want to drink cold tea using a yellow cup," and in other situation "I am hungry" or "I need something to eat" rather than "I want to eat burger placed at right side". The teammate will infer the link between "hot weather", "thirst" and "drink, and "hunger", "food" and "eat". The connection between phrases is that "hot weather" causes "thirst" and it induces a desire to "drink" whereas "hunger" causes the desire to "eat" "food". Consequently, the teammate will offer something to "drink" and other situation will offer something to "eat".

When a robot is working on a team with a person, the robot would be expected to perform comparably to a human teammate. This type of collaboration is presently a challenge in Human-Robot Interaction, although there are several contributions in this direction [1, 84, 95, 111]. Semantic association can be an element to facilitate collaboration when connections between utterances, objects, and skills can be leveraged to communicate incomplete information. The ability to recognize what the user

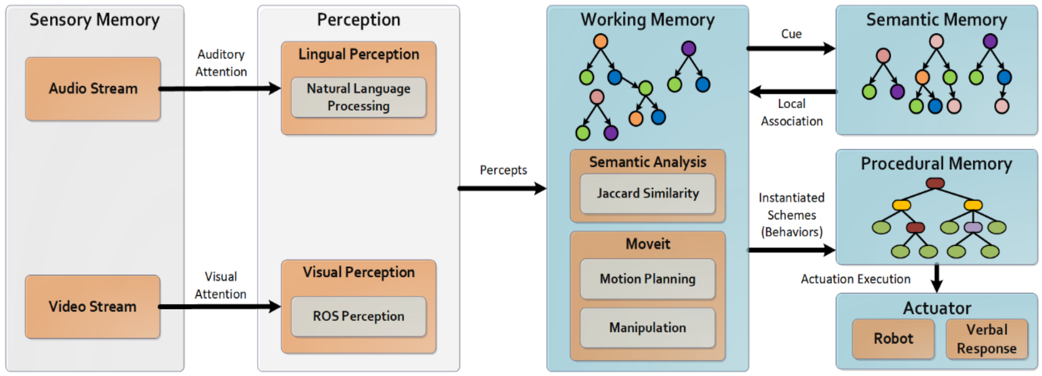
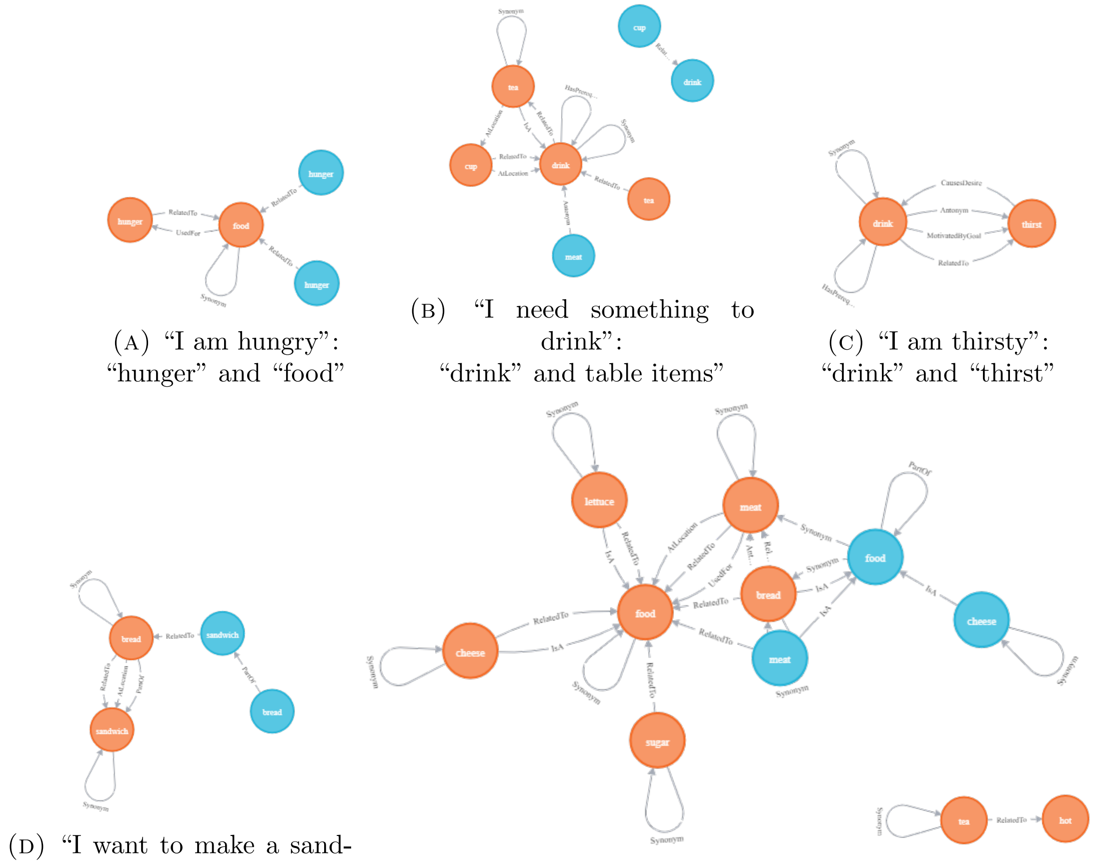


FIGURE 7.2: NiHA 's Minimalistic Cognitive Architecture with Upgraded Perception Layer, Working Memory and Procedural Memory[1]

wants the robot to do next based on a vague or incomplete utterance given a knowledge model of the tasks and objects in the environment [1] can improve the HRI experience. To address these needs, we have developed a methodology based on our existing cognitive [1] and hierarchical control [96] architectures enabling the robots and humans to work collaboratively on joint tasks like "tea making" and "sandwich making". In this regard, we have induced sensory memory, lingual and visual perception, working memory, semantic memory, procedural memory, and actuators (Robot: Baxter, verbal response).

If a robot is given a context and the appropriate objects, it should be able to identify a specific task that will correctly respond to a statement from its partner. Prior work has examined task coordination to encourage users to do different sub-tasks from a robot [96], communicate about task failures [95], and construct new tasks from verbal instruction [89]. If humans and robots can interact verbally about how to accomplish complicated tasks it will emulate another method for HHI. However, such interaction has the added difficulty of teammate communicating with incomplete information or



(A) "I am hungry": "hunger" and "food"

(B) "I need something to drink": "drink" and table items"

(C) "I am thirsty": "drink" and "thirst"

(D) "I want to make a sandwich": "sandwich" and "food"

(E) "I need some food": "food" and table items

(F) "It is hot outside": "hot" and "tea"

FIGURE 7.3: Semantic Graphs extracted from Semantic Memory based on verbal cues

requests that leverages knowledge of the task and the environment.

In this chapter, we propose a solution where the robot can understand the context of the environment in working memory using semantic similarity and can perform the desired task by selecting hierarchical sub-tasks stored in procedural memory. We utilized a cognitive architecture (see Fig 7.2) for task execution in a dynamic environment based on perception and semantic associations. We validate this work using three scenarios by placing the task objects for multiple skills in different positions in front of the robot. In each scenario, the robot is able to understand the context of the environment using ontology by having a dialog conversation with the human. The semantic similarity score enables the robot to decide what skill it needs to perform and executes the skill following the hierarchical task architecture using the objects that falls under the performing skill.

7.1 Problem Formulation

Let the household items presented on table be detected using AR Tags as $I = \{i_1, i_2, i_3, \ldots, i_n\}$.

Let semantic memory of the system be based on k atom of knowledge $S_m = \{Atom_1, Atom_2, Atom_3, \ldots, Atom_k\}$ whereas, $Concept = \{concept_1, concept_2, concept_3, \ldots, concept_k\}$, $Feature = \{feature_1, feature_2, feature_3, \ldots, feature_k\}$, $Relationship = \{relationship_1, relationship_2, relationship_3, \ldots, relationship_k\}$,$Atom = \{(relationship, (concept, concept)), (relationship, (feature, concept)), (relationship, (feature, feature))\}$.

Auditory stimuli are based on m words as,$W_{Auditory} = \{word_1, word_2, word_3, \ldots, word_m\}$. The semantic similarity being evaluated as S be based on Jaccard Similarity Index.

Let the procedural memory be based on Human-Robot Collaborative Architecture having skills as $Skills = \{skill_1, skill_2, skill_3 \ldots, skill_k\}$. The skills are being executed on number of items I with actions as $Actions : Item \rightarrow [pick, place, null]$.

7.2 Methodology

7.2.1 Sensory Memory

Sensory memory is further classified into iconic and echoic memory. The iconic involves the processing of brief images from a video stream whereas the echoic memory processes brief sound bits from an audio stream.

7.2.2 Perception Layer

7.2.2.1 Lingual Perception

The lingual perception is based on the Natural Language Processing (NLP) layer which is further composed of a Part-Of-Speech (POS) tagger, and Tokenization module Tokenization modules tokenizes the spoken commands into words as $W_{Auditory} = \{word_1, word_2, word_3, \ldots, word_m\}$. The tokenized words are further classified into nouns, adjectives and verbs by POS Tagger as noun $N = NN, NNS, NNP, NNPS$, verb $V = \{VB, VBD, VBP, VBN, VBG, VBZ\}$ and adjective $Adj = \{JJ, JJR, JJS\}$.

7.2.2.2 Visual Perception

To detect the objects on the table each object were given different AR tags[109]. AR tags help to identify and track the pose of the object to determine where the object is.

7.2.3 Working Memory

Working Memory (WM) functions as an executive control that is aware of the current situation and can recall earlier events. The basic goal of WM is semantic processing, object grounding, motion planning, and motor command manipulations.

7.2.3.1 Semantic Analysis

The algorithm assesses the semantic similarity between atoms of spoken words and item categories present in the table-top scenario at the time. $W_{Auditory} = \{word_1, word_2, word_3, \ldots, word_m\}$. The semantic function $S : Atom_{W_{Auditory}} \rightarrow Atom_I$. The Similarity Index is being evaluated as

$$S(Atom_{W_{Auditory}}, Atom_I) = max\left(\frac{|Atom_{W_{Auditory}} \cap Atom_I|}{|Atom_{W_{Auditory}}| + |Atom_I| - |Atom_{W_{Auditory}} \cap Atom_I|}\right) \quad (7.1)$$

7.2.4 Semantic Memory

Table 7.1: Semantic Similarity Score between Tagged Words (vertical) and Available Items (horizontal). This information is used to select which objects are most semantically related to words that the partner might say.

	Bread	Cheese	Cup	Lettuce	Meat	Sugar	Tea	Teapot
hot	0.00802	0.00431	0.00493	0.00232	0.00695	0.00656	**0.01163**	0.00116
hungry	0.00063	0.00000	0.00000	0.00000	**0.00345**	0.00000	0.00000	0.00000
thirst	0.00126	0.00000	**0.00578**	0.00524	0.00069	0.00000	0.00513	0.00000
sandwich	**0.02244**	0.01610	0.00655	0.01786	0.01761	0.00269	0.01081	0.00220
drink	0.01008	0.00468	**0.01754**	0.00327	0.00744	0.01547	0.01465	0.00140
food	**0.02879**	0.00906	0.00684	0.00487	0.02537	0.01325	0.00807	0.00024

Semantic memory is developed from WordNet and ConceptNet having 117,659 Synsets (WordNet Nodes), 157,300 Lemma nodes, and 1653804 Concept (ConceptNet) nodes. There are 54 categories of 3730567 relationships [112]. Lemma nodes are the "root words" retrieved from the Concept node and can correlate Concept nodes completely or partially with Synsets whereas an assertion is considered as the atom of knowledge in Semantic Network [113]. The semantic memory is constructed from these atoms of knowledge as concept-relationship-feature or concept-relationship-concept as $Atom = \{(concept, relationship, concept), (concept, relationship, feature)\}$. $S_m = \{Atom_1, Atom_2, Atom_3, \ldots, Atom_k\}$, whereas the $Feature = \{feature_1, feature_2, feature_3, \ldots, feature_k\}$, $Concept = \{concept_1, concept_2, concept_3, \ldots, concept_k\}$ and $Relationship = \{relationship_1, relationship_2, relationship_3, \ldots, relationship_k\}$

Complete details about semantic memory can be accessed at [1]. The depiction subgraphs of semantic memory showing the relationship between the "Food" node and the "Sandwich" node is shown in Fig 7.1.

7.2.5 Procedural Memory

Procedural Memory is Long-Term Memory that is responsible for actions and skills. This memory entirely depends on the type of agent/robot which is being used. We have adopted Human-Robot Collaborative Architecture for the execution of skills i.e., making tea and making a sandwich. Skills describe actions to be taken and their hierarchical constraints.

7.2.5.1 Hierarchical Task Representation

The purpose of the hierarchical task architecture is to enable naturalistic human-robot task execution for complex tasks. This task architecture is based on a complex hierarchical task network in which humans and robots can work simultaneously in the same environment. Almost any real-world task can be divided into sub-tasks and designed as a hierarchical task network. The task can be a collection of sequential, non-sequential, and alternative sub-tasks in the real world.

Our robot control architecture currently enables the system to encode tasks implicating various types of constraints such as sequential (THEN), non-ordering (AND), and alternative paths of execution (OR) [2]. Tasks are represented in a tree structure with two types of nodes: leaf nodes and behavior nodes. Leaf nodes represent tasks to be completed, and behavior nodes represent the hierarchical relationships between those tasks.

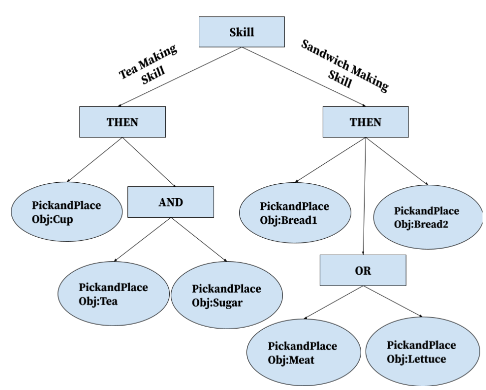


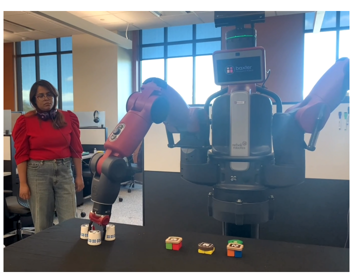
FIGURE 7.4: A new component SKILL was added to the hierarchical task tree which allows the system to choose the skill based on the similarity score. Two types of Skills: 1) Tea Making Skill and 2) Sandwich Making Skill were added under the SKILL node.

For such a hierarchical task, each node in the architecture maintains a state made up of the following components: **1) Activation Level:** a number representing the priority placed by its parent on executing and finalizing a particular 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 indicates that the behavior is active when the node's activation level exceeds a threshold, and **4) Done:** a boolean variable that is set to true when the node has completed its necessary work. The above state information is continuously maintained for each node. The activation spreading technique ensures that the task is executed properly based on constraints by performing both top-down and bottom-up spreading.

(A) Picking Cup from the table

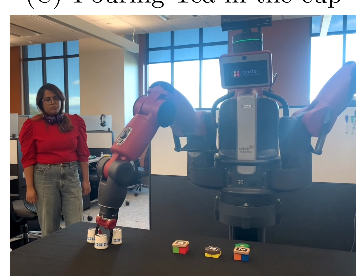
(B) Placing Cup on the table

(C) Pouring Tea in the cup



(D) Placing Tea on the table

(E) Putting Sugar in the cup



(F) Placing Sugar on the table

FIGURE 7.5: The robot is making a cup of tea after the human said, "It is hot outside." The robot determines to execute the Tea Making Skill after analyzing the semantic scores of the available table objects.

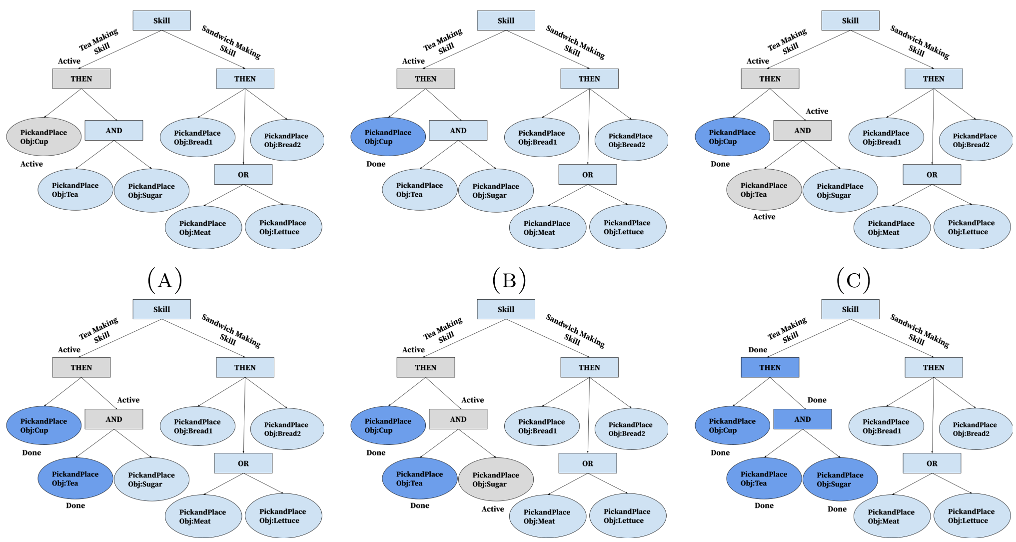
To carry out a task, *activity spreading messages* is sent from the root node to its children to deploy *activity level* throughout the task tree. A bottom-up mechanism distributes *activation potential* throughout the tree by having nodes send their current state as status messages to their parents. In each cycle, a loop assists in maintaining the state of each node in the task structure; the various components of the node's state are checked and updated accordingly.

The controller architecture scales to multiple robots by maintaining a copy of the task tree for each robot. This includes 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.

7.2.5.2 Adding Skill Component in Hierarchical Architecture

In order to extend the previous task architecture, we added a new component **Skill** which decides which of a selection of tasks should be executed based on the environment and interaction. The semantic knowledge module determines the task the robot should complete following the interaction between humans and robots. The skill node can choose the required task for execution by receiving a ROS message in string form. Under the skill node component, we can assign multiple skill tasks to the robot as its child nodes. Whenever the robot chooses a task to perform, it will perform the task accordingly. These skills are designed with constraints like THEN, AND, OR. As shown in Fig 7.4, there are two skills listed under the SKILL node: 1) Tea Making Skill and 2) Sandwich Making Skill. The Skill component determines which task to run based on semantic information and the objects that are available in the environment, the semantic relevance of various objects to words that a user might speak is shown in Table 7.1.

7.3 Experiment Design



(A) (B) (C) (D) (E) (F)

FIGURE 7.6: Order of execution for Tea Making Skill - **((Place Cup)THEN((Place Tea)AND(Place Sugar)))**. After the SKILL node chooses the Tea Making Skill to execute, it only proceeds to execute the nodes that fall under the Tea Making Skill. (a) The THEN node for the Tea Making Skill under the SKILL component is invoked, which initiates the PickAndPlace node for the Cup object, (b) the state of the PickAndPlace of the Cup is changed to Done when the robot finishes placing the cup, (c) AND node gets activated and proceeds to activate the PickAndPlace node for the Tea which makes the robot starts pouring Tea into the cup, (d) the state of the PickAndPlace action of the Tea is changed to Done when the robot finishes pouring Tea and places it on the table, (e) PickAndPlace node for the Sugar under the AND node is activated which makes the robot adding sugar into the cup, (f) PickAndPlace action for the Sugar is Done when the robot places the Sugar on the table, and by doing this the state of all the active nodes under the Tea Making Skill is changed to Done.

To showcase the capabilities of the system we developed, we have designed an interaction using speech conversation between a user and a robot to validate the hierarchical model's capabilities. Based on the user's feedback, the robot can understand the hidden context and perform a skill task utilizing objects from the surrounding environment. We experimented in a lab environment with a human user and a Baxter humanoid robot standing in front of a table with objects. This experiment involves using a robot to make tea and sandwiches. A Kinect v2 camera on top of Baxter's head and Baxter's right-hand camera were used to detect the object's AR tags.

In this human-robot interaction, the robot will decode the tagged word from the human's speech and check the semantic similarity score of the objects(see Table 7.1) associated with the decoded tagged word. The score will help the architecture to decide the appropriate skill task to execute. If the human says a statement like "I am thirsty" or "It is hot outside," the tagged words will be "thirsty" and "hot" respectively. Based on the similarity score, in both cases, it is observed that the objects under the Tea-Making Skill have the highest scores. As a result, the robot will decide to perform the Tea-Making Skill. Based on the task tree (see Fig 7.4), the task will be **((Place Cup)THEN((Place Tea)AND(Place Sugar)))**. According to this task statement, the robot will first pick and place Cup, then pick and place Tea and Sugar in a non-ordered fashion (see Fig 7.5). The hierarchical state representation of each step in executing the Tea-Making Skill is shown in Fig 7.6.

In contrast, if the human says something like "I am hungry", the Meat object from the object list has the highest semantic score associated with the tagged word "hungry" (see Table 7.1). Since Meat falls under the Sandwich-Making skill, the robot will start to make a sandwich. Again, based on the tree, the task will be **((Place Bread1)THEN((Place Meat)OR(Place Lettuce))THEN(Place Bread2))**. Therefore, the robot will pick and place Bread1 then the robot will either pick and place Meat or Lettuce. The robot will then pick and place Bread2 to complete the task.

7.4 Results

In our experiment, the person says, "It is hot outside." Speech recognition provides the ontology with a word string spoken by the user. From the decoded speech, the semantic similarity is being calculated using the Jaccard Similarity index between tagged words and available objects i.e. 'tea', 'sugar', 'cup', 'bread', 'meat', 'cheese', 'lettuce' and 'teapot'(see Table 7.1). The score indicates that the spoken stream is related to 'tea', in this regard, the ontology identified the statement's implied context. The ontology used the available object list and identified the Tea Making task, which has objects that are available and are most connected with the user's speech statement and therefore might be the most helpful. This reflects a connection between the user's statement, the available objects in the environment, and the available tasks that the robot can complete.

Fig 7.6 illustrates the step-by-step state for the tree nodes in our robot architecture for executing the Tea Making skill. At first, when the skill node received the object name ("Tea") which has the highest semantic similarity score (0.01163) among the other objects, the task tree decided to execute the Tea Making Skill from the tree because the object "Tea" falls under this skill. The THEN node was activated for this skill (see Fig 7.6a), and the robot proceeded to pick and place the Cup (see Fig 7.5a Fig 7.5b respectively). When the robot placed the Cup on the table, the status of the Cup node was changed to Done from Active (Fig 7.6b). From the task tree, the robot would activate the AND node (Fig 7.6c) and start picking the tea to pour into the cup (Fig 7.5c). After pouring the tea into the cup, the Tea was set on the table (Fig 7.5d), which made the Tea node in the task tree Done from Active (Fig 7.6d). Then, the robot moved to the next step according to the task tree and activated the Sugar node (Fig 7.6e) and start to put sugar in the cup (Fig 7.5e). At the end, when the Sugar was placed on the table (Fig 7.5f), all the nodes' statuses were changed to Done, and the whole skill task was completed based on the tree design (Fig 7.6f).

We used three validation cases for each skill test. Each time, the objects were in different positions on the table. The robot detected the objects' position every time and performed the expected skill from the hierarchical task tree. Our observations indicated that the robot does not go for objects under different skill sets. Additionally, we provided two statements for each skill test to validate the case scenarios. For instance, we used statements like "I am thirsty" (see Fig7.3c for graph) and "It is hot outside" (see Fig7.3b for graph)for the Tea Making Skill. Likewise, for the Sandwich

Making Skill, we used statements like "I am hungry" (see Fig7.3a for graph) and "I want to make a sandwich" (see Fig7.3d for graph). Furthermore, we have also tried queries "I need some food" (see Fig7.3e for graph) and "I need something to drink" (see Fig7.3f for graph), the respective similarity score about extracted action verbs, nouns and adjectives can be found in Table 7.1.

7.5 Discussion and Future Work

This chapter proposes a way to offer an efficient and flexible human-robot collaboration environment in which the robot teammate can perform the user's desired task by deciphering both vague or clear requests in a natural language form from the human teammate. The ontology played a vital role in the understanding of user commands due to the semantic relationship between various concepts and features. This architecture has the following contributions:

- The system can find an implied link between the context of the situation and the surrounding environment using the ontology approach after interacting with the human user.

- In our extended hierarchical task architecture, the robot will only select the hierarchical sub-tasks that are most relevant to the specific task derived from the ontology approach.

Right now, the robot is performing the skill task after interacting once with a human. However, in the future, we are planning to add more scope to hold conversations to make the system more dynamic and diverse. Rather than a one-line conversation, we plan to hold a series of dialogue conversations between the human and the robot. This will result in a more in-depth understanding of the situation. For example, in the case of the Tea-Making skill task scenario at first, if the human says, “I am thirsty”, then the robot will reply with “Do you want tea?” based on the surrounding object. Later if the human replies with “Yes”, the robot will then proceed to say, “Perfect! Do you want sugar in your tea?”. Based on the human’s reply the system can create a new skill tree which will include sugar or not.

In addition, right now all skill-based sub-trees are manually created in the system by the programmer. However, by applying the idea from Chapter 6, we can make the system more automated. In that case, the tree will be generated by itself which will result in a more flexible, dynamic, and reliable system. In addition, we hope to implement this ontology approach in a multi-human-robot environment to enhance collaboration.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This dissertation aims to answer the problem of a joint task architecture in which humans and robots can collaborate as teammates. This task architecture is based on a complex hierarchical task network in which humans and robots can work concurrently in the corresponding environments. Three types of task synopses can happen in that condition: 1) sequential order, 2) non-ordering, and 3) alternative execution constraints. Teammates are free to pick any sub-task they prefer to complete as long as it serves the constraints policy. The task can be devised as an on-table task. To design this architecture, we extended our previous multi-robot architecture into human-robot architecture. In this system, the robot can recognize a human teammate's intention and determine which sub-task to concentrate on without interrupting the other teammate's sub-task goal. Nevertheless, there can be circumstances when human and robot teammates attempt to achieve the same sub-task. In this state, the system initiated dialogue-based communication between teammates in order to resolve the issue.

The next concern in composing a better human-robot collaboration system is to evaluate the human-robot task architecture. Trustworthiness and trust are some of the most crucial metrics that need to be investigated. People are more inclined to collaborate with a robot if they feel trustworthy toward it. In order to understand the value of trust, we set up an interaction between humans and robots that is homophilic. We examined whether humans and robots can form a homophilic relationship and whether there is a correlation between homophily and trust between humans and robots. This experiment shows the importance of trust in the system.

Another focus of our work is to make the task architecture more flexible and dynamic so that it doesn't need any programmer to specify the task. If the robot can be taught by a human teacher, it will reduce the need to feed the task tree to the robot. We proposed a pipeline where the robot can observe its human teammates working, learn the task design, create/update the task tree, and execute the task. To do this, we designed a task demonstration interface to monitor the human teacher's hand movement using a vision system to understand the task design. After that, we used the previously proposed GA learning framework to learn and construct the task tree. In the end, the robot was able to create and execute the task tree in real-world scenarios. By doing this, the system eliminates the need to manually modify or create the task tree for each revised/updated task design. As a result, people who have less or no programming background can work with robots in daily life.

Ideally, in a collaborative task, the human-robot interaction should resemble a human-human interaction. In that case, the robot should understand what its human teammate needs even if the human teammate is not precise all the time. In short, the robot should have the ability to understand a precise or vague description from the human and can understand the context behind this. If the robot can perform specific tasks based on the context using the objects around it, then the system will be more realistic and reliable for use in daily life. To implement this in our system, the robot and the human had a dialogue interchange between them and based on the human's answer the robot was able to choose a specific skill-based task for this situation using ontology and could execute the task successfully.

The main contributions of this dissertation work are:

1. **Collaborative Human-Robot Hierarchical Task Execution:** Chapter 4 describes the development of a multi-human-robot team based on a multi-robot architecture where tasks with varying environmental conditions are given dynamically. This was accomplished by extending the system's ability to anticipate and accommodate a human's movements. A tree structure represents the human-robot collaborative task as a sequential, non-ordering, and alternative execution path. Using its task representation (e.g., controller), the robot plans

its future actions and tracks the current and future goals of its human teammate. Taking into account the constraints of the defined task as well as the feedback provided by the collaborative partner, the robot determines its next course of action.

- The robot monitors its state and the state of its collaborative human partner.
- The human intention system, designed to supplement our previous robot architecture, continuously sends out messages containing the human intention status of each object. This allows agents to function independently when their tasks do not overlap; however, when agents' goals overlap, a collision occurs on the task tree and dialog is used to resolve the collision. This enables one agent to complete the job while the other moves on to another.

2. **Study on homophily and trust in HRI:** In Chapter 5, we explored homophily between a person and a robot by measuring metrics such as common interest, bonding, and similarity.

   - The purpose of this work is to determine whether similarities between a robot and a person might improve social connection and trust. If such a link exists, then homophily would be an important physical and behavioral design consideration for effective HRI.

- Measured "common interest", "felt bonding", and "trust" between homophilic and non-homophilic conditions.

3. **Efficient task allocation and execution from task demonstration:** In Chapter 6, the robot learned a customized task model by observing the human teacher's movements. By doing this, we were able to reduce the amount of time we devote to manually putting the hierarchical task tree into the system before we run the system. This improved the system's efficiency, flexibility, and dynamic capabilities.

    - Designing and implementing a pipeline where the robot will observe the task demonstrated by the human teacher. It will learn the task design by using the previously proposed learning framework and execute the new task tree by itself.
    - Designing a task demonstration interface by using a vision based system to learn the task demonstrations from the human teacher. The robot will observe the human teacher performing the task while conversing with the human.
    - Enhance the system's ability to learn and execute new tasks without manual specifications.
    - Demonstrating this on a real robot.

4. **Cognitive Approach to Hierarchical Task Selection For Human-Robot Interaction in Dynamic Environments:** Chapter 7 proposed an efficient

and flexible human-robot collaboration environment in which a robot teammate can understand both vague and clear requests from its human partner in natural language and can execute that task in real world scenario.

- Utilizing the ontology approach after interacting with the human user to find implied links between context and the surrounding environment.
- Based on the ontology approach of our extended hierarchical task architecture, the robot will only select the hierarchical sub-tasks that are most relevant for the specific task.

.

8.2 Future Work

8.2.1 Cognitive Approach to Hierarchical Task Selection

According to our proposed system, the robot and human are currently only able to communicate one time to establish the context of the situation. However, in our future work, we are planning to have a more detailed conversation between them to understand the context in depth. This approach will lead the system to create a more specific version of the tree with specific task allocation based on the given answers from the human end. Additionally, we have created skill-based sub-trees to the task tree. On the other hand, using to the concept of Chapter 6, we aim to update the skill tree every time automatically rather than being modified by a programmer, which will make the system more reliable and flexible.

8.2.2 Measuring trustworthiness in a human-robot collaborative task

As part of my dissertation, I studied trust, comfort, and similarity in homophilic relationships (see Chapter 4). However, in the future, I am hoping to work on studies where we can measure metrics like trustworthiness in our proposed human-robot collaborative task and how trustworthiness may affect the task selection process in human-robot interaction.

Bibliography

[1] Syed Tanweer Shah Bukhari and Wajahat Mahmood Qazi. Perceptual and semantic processing in cognitive robots. *Electronics*, 10(18), 2021. ISSN 2079-9292. doi: 10.3390/electronics10182216. URL `https://www.mdpi.com/2079-9292/10/18/2216`.

[2] Janelle Blankenburg, Santosh Balajee Banisetty, Seyed P. Hoseini, Luke Fraser, David Feil-Seifer, Monica Nicolescu, and Mircea Nicolescu. A distributed control architecture for collaborative multi-robot task allocation. In *International Conference on Humanoid Robots*, Birmingham, UK, November 2017. doi: 10.1109/HUMANOIDS.2017.8246931.

[3] Gideon Nave, Colin Camerer, and Michael McCullough. Does oxytocin increase trust in humans? a critical review of research. *Perspectives on Psychological Science*, 10(6):772–789, 2015. doi: 10.1177/1745691615600138. URL `https://doi.org/10.1177/1745691615600138`. PMID: 26581735.

[4] Kristin E. Schaefer, Jessie Y. C. Chen, James L. Szalma, and P. A. Hancock. A meta-analysis of factors influencing the development of trust in automation:

Implications for understanding autonomy in future systems. *Human Factors*, 58 (3):377–400, 2016. doi: 10.1177/0018720816634228. URL https://doi.org/10.1177/0018720816634228. PMID: 27005902.

[5] Guglielmo Papagni and Sabine Koeszegi. Understandable and trustworthy explainable robots: A sensemaking perspective. *Paladyn, Journal of Behavioral Robotics*, 12(1):13–30, 2021.

[6] Manisha Natarajan and Matthew Gombolay. Effects of anthropomorphism and accountability on trust in human robot interaction. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '20, page 33–42, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367462. doi: 10.1145/3319502.3374839. URL https://doi.org/10.1145/3319502.3374839.

[7] Stefan-Dan Ciocirlan, Roxana Agrigoroaie, and Adriana Tapus. Human-robot team: Effects of communication in analyzing trust. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–7, 2019. doi: 10.1109/RO-MAN46459.2019.8956345.

[8] Roya Salek Shahrezaie, Bashira Akter Anima, and David Feil-Seifer. Birds of a feather flock together: A study of status homophily in hri. In Haizhou Li, Shuzhi Sam Ge, Yan Wu, Agnieszka Wykowska, Hongsheng He, Xiaorui Liu, Dongyu Li, and Jairo Perez-Osorio, editors, *Social Robotics*, pages 281–291, Cham, 2021. Springer International Publishing. ISBN 978-3-030-90525-5.

[9] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: theory and practice*. Elsevier, 2004.

[10] Ilche Georgievski and Marco Aiello. Htn planning: Overview, comparison, and beyond. *Artificial Intelligence*, 222:124–156, 2015. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2015.02.002. URL `https://www.sciencedirect.com/science/article/pii/S0004370215000247`.

[11] D. S. Nau, T. C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman. Shop2: An htn planning system. *Journal of Artificial Intelligence Research*, 20:379–404, Dec 2003. ISSN 1076-9757. doi: 10.1613/jair.1141. URL `http://dx.doi.org/10.1613/jair.1141`.

[12] Martin Weser, Dominik Off, and Jianwei Zhang. Htn robot planning in partially observable dynamic environments. In *2010 IEEE International Conference on Robotics and Automation*, pages 1505–1510, 2010. doi: 10.1109/ROBOT.2010.5509770.

[13] Raphaël Lallement, Lavindra de Silva, and Rachid Alami. Hatp: An htn planner for robotics, 2014.

[14] T. Belker, M. Hammel, and J. Hertzberg. Learning to optimize mobile robot navigation based on htn plans. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pages 4136–4141 vol.3, 2003. doi: 10.1109/ROBOT.2003.1242233.

[15] Jun Zhang, Yi Zeng, Huifen Liu, Ying Hu, and Jianwei Zhang. Mbhp: A memory-based method on robot planning under uncertainties. *2011 IEEE International Conference on Robotics and Biomimetics, ROBIO 2011*, 12 2011. doi: 10.1109/ROBIO.2011.6181313.

[16] Raveesh Kandiyil and Yang Gao. A generic domain configurable planner using htn for autonomous multi-agent space system. 2012.

[17] Rafael Cardoso and Rafael BORDINI. A multi-agent extension of a hierarchical task network planning formalism. *ADCAIJ: ADVANCES IN DISTRIBUTED COMPUTING AND ARTIFICIAL INTELLIGENCE JOURNAL*, 6:5, 06 2017. doi: 10.14201/ADCAIJ201762517.

[18] Suying Zeng, Yuancheng Zhu, and Chao Qi. Htn-based multi-robot path planning. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 4719–4723, 2016. doi: 10.1109/CCDC.2016.7531837.

[19] Raphaël Lallement, Lavindra de Silva, and Rachid Alami. HATP: an HTN planner for robotics. *CoRR*, abs/1405.5345, 2014. URL `http://arxiv.org/abs/1405.5345`.

[20] Samir Alili, Rachid Alami, and Vincent Montreuil. *A Task Planner for an Autonomous Social Robot*, pages 335–344. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-00644-9. doi: 10.1007/978-3-642-00644-9_30. URL `https://doi.org/10.1007/978-3-642-00644-9_30`.

[21] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3):143 – 166, 2003. ISSN 0921-8890. doi: https://doi.org/10.1016/S0921-8890(02)00372-X. URL http://www.sciencedirect.com/science/article/pii/S092188900200372X. Socially Interactive Robots.

[22] R.P. Goldman, K.Z. Haigh, D.J. Musliner, and M.J.S. Pelican. Macbeth: a multi-agent constraint-based planner [autonomous agent tactical planner]. In *Proceedings. The 21st Digital Avionics Systems Conference*, volume 2, pages 7E3–7E3, 2002. doi: 10.1109/DASC.2002.1052934.

[23] Ronald C. Arkin, Masahiro Fujita, Tsuyoshi Takagi, and Rika Hasegawa. An ethological and emotional basis for human–robot interaction. *Robotics and Autonomous Systems*, 42(3-4):191–201, Mar 2003. doi: 10.1016/s0921-8890(02)00375-5. URL https://doi.org/10.1016/s0921-8890(02)00375-5.

[24] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000. doi: 10.1177/02783640022067922. URL https://doi.org/10.1177/02783640022067922.

[25] Kerstin Severinson-Eklundh, Anders Green, and Helge Hüttenrauch. Social and collaborative aspects of interaction with a service robot. *Robotics and*

Autonomous Systems, 42(3-4):223–234, mar 2003. doi: 10.1016/s0921-8890(02)00377-9. URL `https://doi.org/10.1016/s0921-8890(02)00377-9`.

[26] M. Hans, B. Graf, and R.D. Schraft. robotic home assistant care-o-bot: past-present-future. IEEE. doi: 10.1109/roman.2002.1045652. URL `https://doi.org/10.1109/roman.2002.1045652`.

[27] R.O. Ambrose, H. Aldridge, R.S. Askew, R.R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark. Robonaut: NASA's space humanoid. *IEEE Intelligent Systems*, 15(4):57–63, jul 2000. doi: 10.1109/5254.867913. URL `https://doi.org/10.1109/5254.867913`.

[28] J. Khurshid and Hong Bing-rong. Military robots - a glimpse from today and tomorrow. In *ICARCV Control, Automation, Robotics and Vision Conference*, 2004. doi: 10.1109/icarcv.2004.1468925. URL `https://doi.org/10.1109/icarcv.2004.1468925`.

[29] J. Casper and R.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. doi: 10.1109/tsmcb.2003.811794. URL `https://doi.org/10.1109/tsmcb.2003.811794`.

[30] Muhammad Awais and Dominik 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. doi: 10.1109/raad.2010.5524605. URL https://doi.org/10.1109/raad.2010.5524605.

[31] Zhichao Wang, Bin Wang, Hong Liu, and Zhaodan Kong. Recurrent convolutional networks based intention recognition for human-robot collaboration tasks. In *International Conference on Systems, Man, and Cybernetics (SMC)*, 2017. doi: 10.1109/smc.2017.8122856. URL https://doi.org/10.1109/smc.2017.8122856.

[32] Yanan Li and Shuzhi Sam Ge. Human–robot collaboration based on motion intention estimation. *IEEE/ASME Transactions on Mechatronics*, 19(3):1007–1014, jun 2014. doi: 10.1109/tmech.2013.2264533. URL https://doi.org/10.1109/tmech.2013.2264533.

[33] Przemyslaw A. Lasota and Julie 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. doi: 10.1177/0018720814565188. URL https://doi.org/10.1177/0018720814565188.

[34] Ren C. Luo and Charly Huang. Human-aware motion planning based on search and sampling approach. In *Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 2016. doi: 10.1109/arso.2016.7736286. URL https://doi.org/10.1109/arso.2016.7736286.

[35] Stefan Escaida Navarro, Maximiliano Marufo, Yitao Ding, Stephan Puls, Dirk Goger, Bjorn Hein, and Heinz Worn. 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. doi: 10.1109/iros.2013.6696495. URL `https://doi.org/10.1109/iros.2013.6696495`.

[36] David Feil-Seifer and Maja 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. doi: 10.1109/DEVLRN.2011.6037331.

[37] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru Asada. Initiative in robot assistance during collaborative task execution. In *International Conference on Human-Robot Interaction (HRI)*, 2016. doi: 10.1109/hri.2016.7451735. URL `https://doi.org/10.1109/hri.2016.7451735`.

[38] Bradley Hayes, Daniel Ullman, Emma Alexander, Caroline Bank, and Brian Scassellati. People help robots who help others, not robots who help themselves. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, aug 2014. doi: 10.1109/roman.2014.6926262. URL `https://doi.org/10.1109/roman.2014.6926262`.

[39] Henny Admoni, Anca Dragan, Siddhartha S. Srinivasa, and Brian Scassellati. Deliberate delays during robot-to-human handovers improve compliance with

gaze communication. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction - HRI '14*. ACM Press, 2014. doi: 10.1145/2559636.2559682. URL `https://doi.org/10.1145/2559636.2559682`.

[40] Chien-Ming Huang, Maya Cakmak, and Bilge Mutlu. Adaptive coordination strategies for human-robot handovers. In *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, jul 2015. doi: 10.15607/rss.2015.xi.031. URL `https://doi.org/10.15607/rss.2015.xi.031`.

[41] K. P. Hawkins, Nam Vo, S. Bansal, and A. F. Bobick. Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 499–506, Oct 2013. doi: 10.1109/HUMANOIDS.2013.7030020.

[42] 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 *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2215–2222, May 2014. doi: 10.1109/ICRA.2014.6907165.

[43] Ruth Schulz, Philipp Kratzer, and Marc Toussaint. Preferred interaction styles for human-robot collaboration vary over tasks with different action types. *Frontiers in Neurorobotics*, 12:36, 2018. ISSN 1662-5218. doi: 10.3389/fnbot.2018.00036. URL `https://www.frontiersin.org/article/10.3389/fnbot.2018.00036`.

[44] Grégoire Milliez, Raphaël Lallement, Michelangelo Fiore, and Rachid 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)*, pages 43–50, 2016. doi: 10.1109/HRI.2016.7451732.

[45] Devin Sandra, Aurélie Clodic, and Rachid Alami. *About Decisions During Human-Robot Shared Plan Achievement: Who Should Act and How?* 10 2017.

[46] Bilge Mutlu, Allison Terrell, and Chien-Ming Huang. Coordination mechanisms in human-robot collaboration. In *Proceedings of the Workshop on Collaborative Manipulation, 8th ACM/IEEE International Conference on Human-Robot Interaction*, pages 1–6. Citeseer, 2013.

[47] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru Asada. Initiative in robot assistance during collaborative task execution. pages 67–74, 03 2016. doi: 10.1109/HRI.2016.7451735.

[48] Ruth Schulz, Philipp Kratzer, and Marc Toussaint. Building a bridge with a robot: A system for collaborative on-table task execution. In *Proceedings of the 5th International Conference on Human Agent Interaction*, HAI '17, page 399–403, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450351133. doi: 10.1145/3125739.3132606. URL `https://doi.org/10.1145/3125739.3132606`.

[49] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001. doi: 10.1146/annurev.soc.27.1.415. URL `https://doi.org/10.1146/annurev.soc.27.1.415`.

[50] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. Exploiting homophily effect for trust prediction. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, page 53–62, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450318693. doi: 10.1145/2433396.2433405. URL `https://doi.org/10.1145/2433396.2433405`.

[51] Aristotle. Rhetoric.nichomachean ethics. *In Aristotle in 23 volumes. Rackman transl. Cambridge: Harvard Univ. Press*, 1934.

[52] Plato. Laws. *Twelve Volumes, Vol. 11. Bury translator. Cambridge: Harvard Univ. Press*, 1968.

[53] Soyoung Jung, Hyoung taek Lim, Sanghun Kwak, and Frank Biocca. Personality and facial expressions in human-robot interaction. In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, pages 161–162. IEEE, July 2012.

[54] Peter H. Kahn, Takayuki Kanda, Hiroshi Ishiguro, Nathan G. Freier, Rachel L. Severson, Brian T. Gill, Jolina H. Ruckert, and Solace Shen. "robovie, you'll have to go into the closet now": Children's social and moral relationships with a

humanoid robot. *Developmental Psychology*, 48(2):303–314, March 2012. ISSN 0012-1649. URL `https://www.learntechlib.org/p/88709`.

[55] Amir Aly and Adriana Tapus. A model for synthesizing a combined verbal and nonverbal behavior based on personality traits in human-robot interaction. *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 325–332, 2013.

[56] Marcel Heerink. Exploring the influence of age, gender, education and computer experience on robot acceptance by older adults. In *Proceedings of the 6th International Conference on Human-robot Interaction*, HRI '11, pages 147–148, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0561-7. doi: 10.1145/1957656.1957704. URL `http://doi.acm.org/10.1145/1957656.1957704`.

[57] Zachary Carlson, Timothy Sweet, Jared Rhizor, Houston Lucas, Jamie Poston, and David Feil-Seifer. Team-building activities for heterogeneous groups of humans and robots. In *International Conference on Social Robotics (ICSR)*, pages 113–123, Paris, France, October 2015. doi: 10.1007/978-3-319-25554-5_12.

[58] Zachary Carlson, Louise Lemmon, MacCallister Higgins, David Frank, Roya Salek Shahrezaie, and David Feil-Seifer. Perceived mistreatment and emotional capability following aggressive treatment of robots and computers. *International Journal of Social Robotics*, 11(5):727–739, Dec 2019. ISSN 1875-4805. doi: 10.1007/s12369-019-00599-8. URL `https://doi.org/10.1007/`

s12369-019-00599-8.

[59] Barbara Kühnlenz and Kolja Kühnlenz. Social bonding increases unsolicited helpfulness towards a bullied robot. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 833–838, 2020. doi: 10.1109/RO-MAN47096.2020.9223454.

[60] Robert Hurley. The decision to trust. 84:55–62, 156, 10 2006.

[61] Nancy R. Buchan, Rachel T.A. Croson, and Sara Solnick. Trust and gender: An examination of behavior and beliefs in the investment game. *Journal of Economic Behavior & Organization*, 68(3):466 – 476, 2008. ISSN 0167-2681. doi: https://doi.org/10.1016/j.jebo.2007.10.006. URL http://www.sciencedirect.com/science/article/pii/S016726810800139X.

[62] Matthias Sutter and Martin G. Kocher. Trust and trustworthiness across different age groups. *Games and Economic Behavior*, 59(2):364 – 382, 2007. ISSN 0899-8256. doi: https://doi.org/10.1016/j.geb.2006.07.006. URL http://www.sciencedirect.com/science/article/pii/S0899825606001199.

[63] Maha Salem, Gabriella Lakatos, Farshid Amirabdollahian, and Kerstin Dautenhahn. Would you trust a (faulty) robot?: Effects of error, task type and personality on human-robot cooperation and trust. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '15, pages 141–148, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-2883-8.

doi: 10.1145/2696454.2696497. URL http://doi.acm.org/10.1145/2696454.2696497.

[64] Peter A. Hancock, Deborah R. Billings, Kristin E. Schaefer, Jessie Y. C. Chen, Ewart J. de Visser, and Raja Parasuraman. A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors*, 53(5):517–527, 2011. doi: 10.1177/0018720811417254. URL https://doi.org/10.1177/0018720811417254. PMID: 22046724.

[65] Sarah Strohkorb Sebo, Margaret Traeger, Malte Jung, and Brian Scassellati. The ripple effects of vulnerability: The effects of a robot's vulnerable behavior on trust in human-robot teams. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, pages 178–186, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-4953-6. doi: 10.1145/3171221.3171275. URL http://doi.acm.org/10.1145/3171221.3171275.

[66] Dana Kulić, Wataru Takano, and Yoshihiko Nakamura. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *The International Journal of Robotics Research*, 27(7):761–784, 2008. doi: 10.1177/0278364908091153. URL https://doi.org/10.1177/0278364908091153.

[67] Aleš Ude, Christopher G. Atkeson, and Marcia Riley. Programming full-body movements for humanoid robots by observation. *Robotics and Autonomous Systems*, 47(2):93–108, 2004. ISSN 0921-8890. doi: https://doi.org/10.

1016/j.robot.2004.03.004. URL https://www.sciencedirect.com/science/article/pii/S0921889004000405. Robot Learning from Demonstration.

[68] Seungsu Kim, ChangHwan Kim, Bumjae You, and Sangrok Oh. Stable whole-body motion generation for humanoid robots to imitate human motions. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2518–2524, 2009. doi: 10.1109/IROS.2009.5354271.

[69] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010. doi: 10.1177/0278364910371999. URL https://doi.org/10.1177/0278364910371999.

[70] Daniel Grollman and Odest Jenkins. Dogged learning for robots. pages 2483 – 2488, 05 2007. doi: 10.1109/ROBOT.2007.363692.

[71] Felix Duvallet and Anthony Stentz. Imitation learning for task allocation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3568–3573. IEEE, 2010.

[72] Bradford A Towle and Monica Nicolescu. An auction behavior-based robotic architecture for service robotics. *Intelligent Service Robotics*, 7(3):157–174, 2014.

[73] Nathan Michael, Michael M Zavlanos, Vijay Kumar, and George J Pappas. Distributed multi-robot task assignment and formation control. In *Robotics*

and Automation, 2008. ICRA 2008. IEEE International Conference on, pages 128–133. IEEE, 2008.

[74] M Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[75] Anahita Mohseni-Kabir, Sonia Chernova, and Charles Rich. Collaborative learning of hierarchical task networks from demonstration and instruction. In *RSS Workshop on Human-Robot Collaboration for Industrial Manufacturing, Berkeley, CA*, 2014.

[76] Anahita Mohseni-Kabir, Charles Rich, Sonia Chernova, Candace L Sidner, and Daniel Miller. Interactive hierarchical task learning from a single demonstration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 205–212, 2015.

[77] Bradley Hayes and Brian Scassellati. Autonomously constructing hierarchical task networks for planning and human-robot collaboration. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 5469–5476. IEEE, 2016.

[78] Negin Nejati, Pat Langley, and Tolga Konik. Learning hierarchical task networks by observation. In *Proceedings of the 23rd international conference on Machine learning*, pages 665–672, 2006.

[79] Yichi Zhang and Joyce Chai. Hierarchical task learning from language instructions with unified transformers and self-monitoring, 2021. URL `https://arxiv.org/abs/2106.03427`.

[80] Ross A Knepper, Dishaan Ahuja, Geoffrey Lalonde, and Daniela Rus. Distributed assembly with and/or graphs. In *Workshop on AI Robotics at the Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[81] Kalpesh Padia, Kaveen Herath Bandara, and Christopher G. Healey. A system for generating storyline visualizations using hierarchical task network planning. *Computers & Graphics*, 78:64–75, 2019. ISSN 0097-8493. doi: https://doi.org/10.1016/j.cag.2018.11.004. URL `https://www.sciencedirect.com/science/article/pii/S0097849318301821`.

[82] Dian Liu, Hongwei Wang, and Hui Lu. Composition of construction services with hierarchical planning on digital platform. *Automation in Construction*, 141:104449, 2022. ISSN 0926-5805. doi: https://doi.org/10.1016/j.autcon.2022.104449. URL `https://www.sciencedirect.com/science/article/pii/S0926580522003223`.

[83] Franziska Doris Wolf and Ruth Stock-Homburg. Human-robot teams: A review. In Alan R. Wagner, David Feil-Seifer, Kerstin S. Haring, Silvia Rossi, Thomas Williams, Hongsheng He, and Shuzhi Sam Ge, editors, *Social Robotics*, pages 246–258, Cham, 2020. Springer International Publishing. ISBN 978-3-030-62056-1.

[84] Jinpeng Mi, Song Tang, Zhen Deng, Michael Goerner, and Jianwei Zhang. Object affordance based multimodal fusion for natural human-robot interaction. *Cognitive Systems Research*, 54:128–137, 2019. ISSN 1389-0417. doi: https://doi.org/10.1016/j.cogsys.2018.12.010. URL `https://www.sciencedirect.com/science/article/pii/S1389041718302018`.

[85] Guido Bugmann, Stanislao Lauria, Theocharis Kyriacou, Ewan Klein, Johan Bos, and Kenny Coventry. Using verbal instructions for route learning: Instruction analysis. *Proc. of Torwards Intelligent Mobile Robots Conf., Manchester, UK*, 01 2001.

[86] Harel Yedidsion, Jacqueline Deans, Connor Sheehan, Mahathi Chillara, Justin Hart, Peter Stone, and Raymond J. Mooney. Optimal use of verbal instructions for multi-robot human navigation guidance. In Miguel A. Salichs, Shuzhi Sam Ge, Emilia Ivanova Barakova, John-John Cabibihan, Alan R. Wagner, Álvaro Castro-González, and Hongsheng He, editors, *Social Robotics*, pages 133–143, Cham, 2019. Springer International Publishing. ISBN 978-3-030-35888-4.

[87] Cetin Mericli, S.D. Klee, J. Paparian, and M. Veloso. An interactive approach for situated task teaching through verbal instructions. *AAAI Workshop - Technical Report*, pages 47–52, 01 2013.

[88] L. Villamar Gómez and J. Miura. Ontology-based knowledge management with verbal interaction for command interpretation and execution by home service

robots. *Robotics and Autonomous Systems*, 140:103763, 2021. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2021.103763.

[89] Monica Nicolescu, Natalie Arnold, Janelle Blankenburg, David Feil-Seifer, Santosh Balajee Banisetty, Mircea Nicolescu, Andrew Palmer, and Thor Monteverde. Learning of complex-structured tasks from verbal instruction. In *International Conference on Humanoid Robots*, Toronto, Canada, October 2019.

[90] Roya Salek Shahrezaie, Lotey B. Manalo, Aaron Brantley, Casey Lynch, and David Feil-Seifer. Advancing socially-aware navigation for public spaces. In *International Conference on Robot and Human Interactive Communication (RO-MAN)*, Naples, Italy, August 2022.

[91] Francisco Cruz, Sven Magg, Cornelius Weber, and Stefan Wermter. Training agents with interactive reinforcement learning and contextual affordances. *IEEE Transactions on Cognitive and Developmental Systems*, 8(4):271–284, 2016. doi: 10.1109/TCDS.2016.2543839.

[92] Yohan Breux, Sebastien Druon, and Rene Zapata. From perception to semantics: An environment representation model based on human-robot interactions. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 672–677, 2018. doi: 10.1109/ROMAN.2018.8525527.

[93] Miguel Oliveira, Luís Seabra Lopes, Gi Hyun Lim, S. Hamidreza Kasaei, Ana Maria Tomé, and Aneesh Chauhan. 3d object perception and perceptual learning in the race project. *Robotics and Autonomous Systems*, 75:614–626, 2016. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2015.09.019. URL `https://www.sciencedirect.com/science/article/pii/S0921889015002146`.

[94] Alexandre Antunes, Lorenzo Jamone, Giovanni Saponaro, Alexandre Bernardino, and Rodrigo Ventura. From human instructions to robot actions: Formulation of goals, affordances and probabilistic planning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5449–5454, 2016. doi: 10.1109/ICRA.2016.7487757.

[95] Janelle Blankenburg, Mariya Zagainova, Stephen M. Simmons, Gabrielle Talavera, Monica Nicolescu, and David Feil-Seifer. Human-robot collaboration and dialogue for fault recovery on hierarchical tasks. In *International Conference on Social Robotics (ICSR)*, CO, October 2020. doi: 10.1007/978-3-030-62056-1_13.

[96] Bashira Akter Anima, Janelle Blankenburg, Mariya Zagainova, Seyed (Pourya) Hoseini, Muhammed Tawfiq Chowdhury, David Feil-Seifer, Monica Nicolescu, and Mircea Nicolescu. Collaborative human-robot hierarchical task execution with an activation spreading architecture. In *International Conference on Social Robotics*, pages 301–310, Madrid, Spain, November 2019. doi: https://doi.org/10.1007/978-3-030-35888-4_28.

[97] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. 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.

[98] S. Pourya Hoseini A., M. Nicolescu, and M. Nicolescu. 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. doi: 10.1109/COGSIMA.2018.8423982.

[99] Paul Lazarsfeld and Robert Merton. Friendship as social process: A substantive and methodological analysis. pages 18–66, 1954.

[100] Mic Bowman, Saumya K. Debray, and Larry L. Peterson. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.*, 15(5):795–825, November 1993.

[101] E. P. Bernier and B. Scassellati. The similarity-attraction effect in human-robot interaction. In *2010 IEEE 9th International Conference on Development and Learning*, pages 286–290, Aug 2010. doi: 10.1109/DEVLRN.2010.5578828.

[102] James Moody. Race, school integration, and friendship segregation in america. *American Journal of Sociology*, 107(3):679–716, 2001. ISSN 00029602, 15375390. URL `http://www.jstor.org/stable/10.1086/338954`.

[103] U.S. Census Bureau. Race and ethnicity. 2017. URL `https://www.census.gov/mso/www/training/pdf/race-ethnicity-onepager.pdf`.

[104] Linda L. McCroskey, James C. McCroskey, and Virginia P. Richmond. Analysis and improvement of the measurement of interpersonal attraction and homophily. *Communication Quarterly*, 54(1):1–31, 2006. doi: 10.1080/01463370500270322. URL `https://www.tandfonline.com/doi/abs/10.1080/01463370500270322`.

[105] Jiun-Yin Jian, Ann Bisantz, and Colin Drury. Foundations for an empirically determined scale of trust in automated systems. 4:53–71, 03 2000.

[106] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1): 415–444, 2001. doi: 10.1146/annurev.soc.27.1.415. URL `http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.soc.27.1.415`.

[107] Janelle Blankenburg, Santosh Balajee Banisetty, Seyed Pourya Hoseini Alinodehi, Luke Fraser, David Feil-Seifer, Monica N. Nicolescu, and Mircea Nicolescu. A distributed control architecture for collaborative multi-robot task allocation. In *17th IEEE-RAS International Conference on Humanoid Robotics, Humanoids 2017, Birmingham, United Kingdom, November 15-17, 2017*, pages 585–592, 2017. doi: 10.1109/HUMANOIDS.2017.8246931. URL `https://doi.org/10.1109/HUMANOIDS.2017.8246931`.

[108] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

[109] Scott Niekum. ar_track_alvar. `https://github.com/ros-perception/ar_track_alvar`, 2016.

[110] Vahid Mokhtari, Roman Manevich, Luís Seabra Lopes, and Armando J. Pinho. Learning the scope of applicability for task planning knowledge in experience-based planning domains. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3973–3979, 2019. doi: 10.1109/IROS40897.2019.8968013.

[111] Manuel Lopes, Francisco S. Melo, and Luis Montesano. Affordance-based imitation learning in robots. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1015–1021, 2007. doi: 10.1109/IROS.2007.4399517.

[112] stsbukhari. stsbukhari/wordnet-conceptnet-neo4j. `https://github.com/stsbukhari/wordnet-conceptnet-neo4j`, 2022.

[113] John F Sowa. *Semantic Networks*. John Wiley & Sons, Ltd, 2006. ISBN 9780470018866. doi: https://doi.org/10.1002/0470018860.s00065. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/0470018860.s00065`.