POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Effects of dynamic planning on the adaptability of a cobotic system to human constraints for a cooperative HRI assembly task

Supervisor

Prof. Marina Indri

Candidate

Beatrice Piras

Supervisors at LIG lab - Grenoble UGA

Prof. Damien Pellier
Prof. Humbert Fiorino

October 2023

Abstract

The increasing presence of robots in industrial settings has led to a growing interest in human-robot collaboration (HRC). The focus of this thesis is on the problem of autonomous and automatic task planning for adapting a cobotic system's plan of actions to specific human factors and constraints. To favor ergonomic choices and reduce muscoloscheletal disorders, the human should have the freedom of choice of his next action. To increase security and reduce mental load, the robot needs to be aware of human constraints and plan also according to them. For this the robot needs to dynamically plan its next actions taking into consideration both factors: human choices and human limits. We propose a framework based on the use of AI planning and PDDL for the dynamic update of the cobot's plan, in the context of a human robot collaboration assembly task. An experiment is conducted to validate the proposed implementations and assess their usefulness. The experiment involves a collaborative task between humans and robots, evaluating performance, risk handling, and acceptability. The expected results aim to demonstrate the adaptability of the cobotic system, improved human experience, and effective planning.

Acknowledgements

First of all, I am so grateful to have been able to receive a good education from two top universities, Politecnico di Torino in Italy and in Grenoble INP-Ensimag in France.
I would like to thank my supervisor in Politecnico di Torino, Marina Indri, and my supervisors in Grenoble, Damien Pellier and Humbert Fiorino at the LIG lab. Thank you for all the support in these months of internship and for their help in reviewing my manuscript.
A special mention goes to my colleagues Belal Hmedan and Étienne Fournier that helped me so much in these months of my internship.
I want to thank my colleagues and friends from the LIG lab, we supported each other through the hardships of every day research, through numerous lunches and coffee breaks, you all made my days better. In particular, Fatima, Rouba and Dani: thank you for being some of my favourite comfort human beings for these past 6 months, i appreciate you and cherish out friendships a lot.

To the people that I met in my PoliTo years or that i left back in Sardegna and has always been my safety net: Grazie. You all know how much I love you, but maybe you don't know how much you helped me. In particolare Marghe, Melis, Sam, Omar, Teo, le mie fringuelle Marty e Ila, thank you for being in my life and being my small beams of light.
Marghe, la mia dœnna, grazie per essere quella persona che mi ha sempre supportato con tutta la gentilezza e comprensione ed avermi sempre fatto sentire validated. Da quando ci conosciamo, sei sempre stata un mio punto di riferimento costante in ogni fase della nostra vita e tra tutti i cambiamenti che abbiamo attraversato e che attraverseremo in futuro, you are my support system and i am soso thankful to have a marghes in my life.
Melis, mela, there is a little bee shaped place in my heart for you. Thank you for always being by my side through ups and downs, through crisis and moments of joy, past and future. With your unconditional love that you give (not only to me but to everyone around you), you constantly remind me to be true to myself because thats just enough to be appreciated. Ad meliora.
Omar, il mio compagno di studi per eccellenza. I am so glad our paths have crossed and keep crossing even if we dont plan for it. The universe has our backs. Mi hai resa una studentessa migliore, sempre motivata a fare meglio. Sei il mio compagno di obbiettivi e aspirazioni e spero che avremo ancora questa influenza positiva l'uno sull'altro anche con l'inizio di questo nuovo capitolo nelle nostre vite, per ora we must gogo, go our separate ways.

Infine, il piú grande dei ringraziamenti va alla mia famiglia che mi ha sempre

sostenuto nelle mie scelte e supportato in ogni modo. Mi avete sempre trasmesso a vostro modo tutto quello di cui avevo bisogno per la vita. Mi sento davvero fortunata ad avere un padre che mi ha insegnato ad avere curiosità verso le cose nuove e che mi ha trasmesso una mentalità logica che adoro, e sono stata fortunata ad avere una madre che mi ha insegnato la cura dei dettagli, nel "leggi bene tutto e non farti fregare", nel fare le cose come si deve ma se qualcosa va male continuare a persistere senza arrendersi.

Ma alla fine dei conti, questa tesi la dedico a mia nonna, nonna Lina, che mi continua ad insegnare tanto su come avere classe, su come essere umili nella propria vita anche facendo grandi cose e facendole bene, ma in silenzio. Non mi stancherò mai di parlare con te. Ogni volta mi insegni a mettere in discussione tutto e che non è mai troppo tardi per fare domande e imparare cose nuove anche "po una beccixedda cummenti 'e mei".
PS. mia nonna è stata l'unica (a 92 anni suonati e con una onorevolissima seconda elementare) ad aver insistito per sapere su cos'era la tesi e ad insistere per spiegargliela in termini semplici, mentre mi faceva domande su domande, con la curiosità di una bambina. E questo già riassume il perchè la adoro e questa tesi non può che essere dedicata a lei.
Grazie.

Beatrice

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The presence of robots in industrial settings is nowadays an affirmed and accepted phenomenon [1]. Their usefulness is undoubted, since they are able to replace human labor for dangerous or repetitive simple manual tasks. While this is true for relatively simple automated tasks, human assistance is still required for more complicated, variable or delicate manipulation.
For this reason, in recent years, there has been a growing interest in the collaboration between humans and robots in the industry[2]. HRC offers significant benefits, such as more flexible task completion, increased productivity, and the ability to combine the resilience of robots with the critical thinking and experience of humans[3] [4]. There are still many obstacles that need to be addressed to make human-robot collaboration a reality.
This thesis tries to analyze the problems by identifying the main challenges that hinder the implementation of human-robot collaboration in the industry. Specifically, this research aims to find an appropriate solution to the problem of automatic task planning for the adaptation of a cobotic system to specific human factors. Moreover, an important objective of this thesis is that of organizing an experiment to prove the usefulness of the implementations proposed.

The research behind this thesis was conducted during an internship in the Laboratoire d'informatique Grenoble -UGA, carried out during an Erasmus Exchange project in Grenoble INP-Ensimag. The research is financed by the Cross Disciplinary Program BOOT. The experiment was carried out in collaboration with a Work Psychology expert and PHD student as part of his PHD research. The starting robot setup was created by the MARVIN team and included a well functioning version of an initial framework. The additions and modifications made to the framework were all well adjusted and thought through to fit the realization of a new experimental protocol, that was conceived by the author of this thesis and approved by the psychology expert and supervisors of the project. The main

interventions made to the framework are in the Decision Making system and all the databases of initial block positions. Minor modifications were made to the vision and execution system and to the ros code of the yumi bot, major bugs were fixed and the system has been overall optimized during the internship months.
The internship lasted 5 months. The first month was dedicated to the state of the art research and the idealization of the experimental protocol. The second month was dedicated to

- getting acquainted with the robot and the framework, learning the PDDL language and the use of AI solvers
- Modifying the framework to fit the experimental design and fix major bugs in the system

During the third month

- the simulations of the system and pre-tests with some human subjects were carried out
- the participants were recruited and the experiments were conducted that lasted 2 weeks

During the fourth month the writing of the thesis was formally started and the different data collected in the experiment was analyzed and some conclusions were drawn.
During the fifth month an initial report was fully written, we carried out a demo presenting the robot and the framework for the industry and for some university partners, and a presentation of the system and the experiment was made for the BOOT project. At the end of the month the defence of the thesis for Grenoble INP- Ensimag was presented in the context of my Erasmus exchange.

1.1 Context

1.1.1 Human robot collaboration and Cobots

The concept of human robot collaboration and in particular the concept of "cobots" was introduced by Peshkin et al in [5] a quarter of a century ago.
A cobot is a robot that is able to work in close contact and collaboration with a human partner to achieve a common goal. They allow the human to concentrate on the tasks that might require specific human capabilities while taking care of other actions that might be unfeasible for the human or simply easier for a robot[4]. The collaboration between the two partners ideally helps the management of human mental load or physical fatigue, and potentially reduce the risk of accidents in the work environment.

Reasons behind the lack of Human-Robot Collaboration

Despite the potential benefits of HRC, its use in the industry is still very sporadic. The main reason for this is that adapting robots to work with humans is not a straightforward problem. From a human point of view, robots are historically too dangerous to work with in proximity [6], less trustworthy to collaborate with than another human, expensive to maintain and complicated to adapt to different types of tasks, and unsuitable to manage human unpredictability[4].

When considering regulations at the governmental and international level, it is difficult to find well defined protocols and ethical policies to follow regarding Robotics. Europe is leading the way in defining norms and regulation for collaborative industrial robots[7]. Some of the most important norms to follow are about the security of the individual that can be in contact with the robot.
When thinking about Robot regulation, Asimov's Three laws of Robotics (Asimov, 1950) come to mind:

- A robot may not injure a human being or, through inaction, allow a human being to come to harm
- A robot must obey the orders given it by human beings except where such orders would conflict with the First Law
- A robot must protect its own existence as long as such protection does not conflict with the first or second law

Although they offer a starting base, their purpose was not that of regulating real life existing robots, but fictional characters. Real robots are much more complicated to regulate and especially when intelligent robots come to play [8]. Mainly, industrial robots should not bring physical or psychological arm to humans and should be compliant with laws that apply to any other industrial tool[7] [8].

From the point of view of effectiveness, the implementation of HRC in the industry requires a certain level of autonomy from the robot that is not yet reached, as well as a strategy to communicate its intentions to its human partner.
Considering these last aspects, there are still many obstacles to overcome, such as:

- secure body collision avoidance [9] [10]
- Ergonomics [6][11]
- efficient action planning [6]
- adaptability to human constraints and mental load [6]

- general perceived usefulness and trust [4]

As a main concern, robots are not yet efficient in realizing dynamic tasks. Ideally the cobot should be able to adapt to the actions that the human chooses to take while ensuring that it does not repeat those actions and can modify its planning accordingly. It is crucial to address these challenges to make human-robot collaboration feasible.

Human Choice and Robot Autonomy in HRC

For human-robot collaboration to be effective, it is crucial to allow the human to make decisions and have the freedom to choose the best course of action while freeing them from the burden of having to communicate every time they take a decision.
The robot must have a certain level of autonomy, and a strategy to communicate its intentions to its human partner must be established. The ideal scenario is one where the human is not burdened by having to manage the robot's actions and can focus on their own tasks.
The robot must also be able to plan its actions based on the human's decisions automatically detected, but to be able to know the state of the environment, the system requires a vision unit connected to the planning unit that can be updated semi-automatically and quickly.
To ensure the human is aware of the robot status and to give more autonomy to both partners, a communication channel between the human and the robot is deemed necessary.

Consideration of Human Constraints

When implementing human-robot collaboration, it is essential to consider human constraints, such as security concerns or the difficulty of performing certain actions. The robot should be able to recognize when the human is experiencing a constraint and decide whether it would be better to assign the task to the robot or allow the human to decide if they are up for it. It is crucial to give the human the freedom of choice in their actions to be able to fully exploit their benefits.

Adaptive Dynamic AI planning for HRC

Dynamic AI planning is necessary to achieve flexible human-robot collaboration. PDDL planning in recent years has made good improvements in task planning domains. A good static planning alone is not enough, since what the robot needs is a dynamic way to change the planning to respond to the human decisions.
For that, it is essential to have a way of always planning in the best way knowing

the state of the environment and change the planning accordingly when necessary.

1.1.2 Automated planning

Automated planning is a branch of Artificial Intelligence that focuses on the development of techniques that allow the creation of a plan. Starting from an initial state, formulates a series of actions considering prerequisites and/or constraints to reach a preset result or end goal . The STRIPS planner [12] was introduced in 1971 as one of the first formal modelling languages to represent planning problems that was used to plan for an autonomous robot movements. Nowadays, Automated Planning problems are commonly expressed thought PDDL (Planning Domain Definition Language) [13] as it grew to become the standard language for AI planning problems and domains.

1.2 Contribution

The focus of this thesis comes from the necessity of solving two main problems in the context of HRC:

1. The human should have the freedom to act as they need and want while the robot should be able to adapt in real time to their choice of actions.
2. The robot should be aware of human constraints and adapt its planning of tasks accordingly to reduce the human mental load or/and ensure their physical safety.

A secondary necessity is that of making the human feel at ease and helped in collaborating with the robot.

The freedom of choice of the human in HRC is a necessity that stems from the ergonomic problems [11] that can arise when a worker is given repetitive tasks over a long period of time. The developement of work-related musculoskeletal disorders (MSDs)[14] is strictly correlated to this problem.
Recent studies show how reconfigurable human-robot collaboration (HRC) frameworks can help limit these consequences[15][16].
A focus of this paper is then to allow the human to be able to make more natural and ergonomic choices and not limit the range of decisions in an attempt to lower the risk of MSDs in workers.

In this thesis we will propose a framework that wants to address these problems and we will explore the impacts of the adaptive cobotic system proposed through an experimentation.

1.2.1 The framework

The objective of the HRC system developed is that of handling unexpected human behaviors and constraints through Automated Planning, modifying dynamically the task planning and robot decisions. We are gonna then experimentally evaluate the performance through standard criteria.

As a solution, the proposition is that of implementing:

- **Dynamic adaptive Planner** based on PDDL: the plan changes according to the human behavior and the handling of the constraints.
- **Perception system**: the actions of the human are automatically perceived through the changes in the environment setup.
- **Graphic user interface** to communicate: The human is aware of the robot perception of the environment and is free to decide when to activate the robot for help. The robot will communicate its decided action before performing it, to increase acceptability and avoid repetition of the actions.

1.2.2 The Experiment

To evaluate the system's functionality an experiment will be formulated.
The experiment will be aimed at the exploration of the impacts of the realized adaptive cobotic system in a constrained work environment and human variabilities, and its effects on workload, efficiency, and acceptability of the human.

The experiment will be based on the realization of a collaborative task between human and robot, where the two actors will have to manipulate a stock of Lego Duplo blocks in order to complete a goal configuration that is known to both.

To measure performance, we will consider the number of errors made, the completion time of the task, and the number of risks taken by the human, together witht he delay on taking these risks. Workload will be assessed using the NASA TLX method, which measures perceived workload of the participant. The number of gestures made by the participants' upper limbs within the work zone will also be recorded. Finally, acceptability will be evaluated through a questionnaire assessing perceived usability, coherence, pleasure, usefulness, social influence, and trust.

The human variability

Two main human variabilities will be accounted for in the experiment:

- **Cognitive variability**: the differences that every human has in their cognitive reasoning. The freedom of choice and unpredictability of human behavior.
- **Expertise level**: Half of the participants were thought the complex task beforehand and they assembled it alone to become acquainted with the task, the other half realized the complex task for the first time directly with the help of the robot.

The constraint

A constraint will be introduced for the human, that the robot will manage through specific planning. We simulate a chemical work environment where the human and robot need to handle some dangerous or less dangerous chemical substances, represented by different color duplo blocks.
The substances can have two levels of danger: regular and toxic. For handling regular substances, the use of surgical latex gloves is enough. For the manipulation of a toxic substance the use of special gloves is required. The special glove cannot be used to manipulate a regular substance because it is considered to be contaminated.

1.3 Expected results

The results of the experiments will hopefully show that the cobotic system was able to smoothly adapt to the chosen human actions while offering the appropriate support through the robot manipulation.
The humans would hopefully feel helped and at ease in the collaboration with the robot partner and their mental load and/or perceived safety should show improvements.
The planning carried out by the robot should adapt smoothly to the different human choices caused by the cognitive variabilities of each human and also should adapt to the eventual pre-declared constraints in attempt to help the human.

1.3.1 Structure of the thesis

Chapter 2 will explore the current state of the art concerning the field of interest of this paper. Chapter 3 will describe the framework proposed and the contributions given by the author of this thesis. Chapter 4 will focus on the details of the experiment and chapter 5 on its results.Finally, chapter 6 will deline the conclusions that can be carried out from the collected data. The thesis will end with an analysis of future necessary work in the field and finally present eventual open issues with the research.

Chapter 2

State of the Art

For this thesis's focus, the problem at hand is inherently multidisciplinary, requiring the understanding of various subject areas to achieve comprehensive and effective results. Consequently, in order to provide a thorough analysis of the current advancements, the state of the art will be divided into distinct thematic sections. Specifically, this review will focus on two primary domains:

- The application of PDDL for task planning;
- AI Planning for cobotic systems, specifically which has been employed in related experiments and their characteristics;

By examining these key fields, a comprehensive understanding of the existing research landscape can be established.

2.1 Planning methodologies for Cobots

2.1.1 What is PDDL

Planning Domain Definition Language (PDDL), presented in [13], is a formal language and framework used for modeling and solving planning problems in the field of automated planning and artificial intelligence. PDDL provides a standardized and expressive representation to define the components of a planning problem, including the initial state, actions, goals, actors and possible constraints. PDDL serves as a powerful tool for both researchers and practitioners in the planning domain. It offers a structured and formalized approach to representing planning problems. Furthermore, planners can precisely specify the initial state of a system, the actions available to manipulate the state, and the desired goal state. This enables the automated generation of plans or schedules that achieve

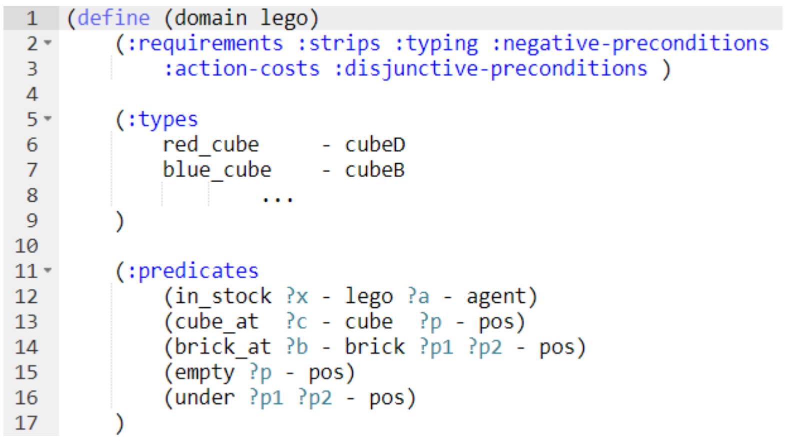
the specified goals while adhering to any constraints or resource limitations.
Using PDDL has several other advantages, some of these are:

- It promotes clarity and precision in problem specification, ensuring a shared understanding among its users.
- It allows for the standard representation of planning problems, enabling the reuse of domain-specific notions and promoting the development of reusable planning models.
- It supports the integration of different planning techniques and algorithms, allowing planners to select and apply the most suitable approach for their specific problem.
- It is still being implemented and updated yearly. More precisely, there is active ongoing research on the creation of more efficient and more personalized AI solvers.

How does PDDL work

In order to illustrate the functionality of PDDL we will use some snippets of code from our own implementation.

Domain Definition



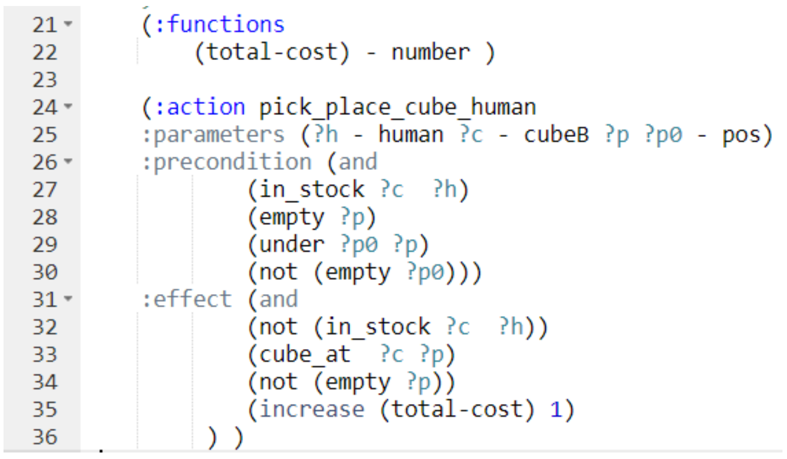
```
 1  (define (domain lego)
 2      (:requirements :strips :typing :negative-preconditions
 3          :action-costs :disjunctive-preconditions )
 4
 5      (:types
 6          red_cube      - cubeD
 7          blue_cube     - cubeB
 8                  ...
 9      )
10
11      (:predicates
12          (in_stock ?x - lego ?a - agent)
13          (cube_at  ?c - cube  ?p - pos)
14          (brick_at ?b - brick ?p1 ?p2 - pos)
15          (empty ?p - pos)
16          (under ?p1 ?p2 - pos)
17      )
```

Figure 2.1: Domain definition in PDDL

The definition of a Domain is always essential when using PDDL. A *domain* is defined in figure 2.1 line 1 attributing the name "lego" to the domain. Then, it is essential to define the *requirements*, that allow the representation of the domain.

After that, some *types* are defined. They represent real objects in the world that will be used in the domain description when declaring actions or preconditions. The *predicates* are instead some facts or states that the objects can have or be in. They are useful for precondition declarations and effects caused by action definitions.

In figure 2.2 we defined a function and an action. *Functions* are used to represent real-valued quantities or numeric attributes associated with the states in a planning problem, in this case, a function is needed to keep track of the accumulated total cost of the actions.
Actions represent the fundamental building blocks for defining the behavior and transformations in a planning domain. They describe the possible actions or operations that can be performed by actors to manipulate the state of the world. In this example, the action representing the pick and place of a cube made by the human is described.
The parameters are defined: a human, a cubeB and a pos will be the types involved in this action.
As we can see, some *preconditions* need to be respected for the human to be able to be assigned the action by the planner. For example some of the preconditions described are: the cube needs to be in the stock of the human, the arrival position for the cube needs to be empty and it needs to have another object or the ground underneath it(the cube cannot be placed mid air).
Also some *effects* of the action can be described. Effects are modifications to the state of the world that happen because the action has been performed. Some effects of the action in our example are: the cube manipulates is not in the human stock anymore, the cube is placed at the new position, the position is not empty anymore, the total cost of actions increases by 1 unit.



```
(:functions
    (total-cost) - number )

(:action pick_place_cube_human
:parameters (?h - human ?c - cubeB ?p ?p0 - pos)
:precondition (and
        (in_stock ?c  ?h)
        (empty ?p)
        (under ?p0 ?p)
        (not (empty ?p0)))
:effect (and
        (not (in_stock ?c  ?h))
        (cube_at  ?c ?p)
        (not (empty ?p))
        (increase (total-cost) 1)
    ) )
```

Figure 2.2: Functions and Actions in PDDL

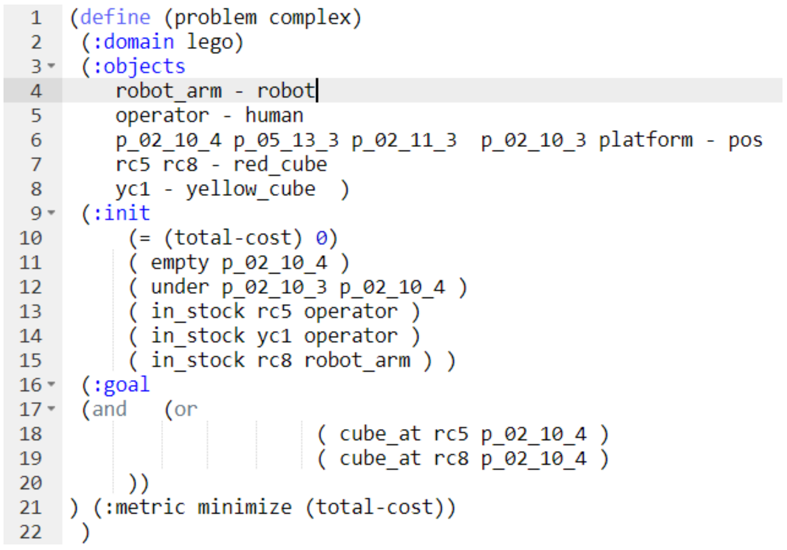
Problem definition

The second essential part of a PDDL planning task definition is the Problem definition. In PDDL, a problem represents a specific scenario within a defined planning domain. It defines the initial state of the world and the desired goal state that the planner aims to achieve through a sequence of actions defined in the domain.

The problem defined in figure 2.3 shows the basic structure of a problem that refers to the domain "lego" previously encountered.
Firstly, the *objects* involved in the problem are listed. In this example we find a robot arm, an operator, various positions and blocks.
The problem continues with the initialization of the current state. The initial state of the problem describes the starting conditions or state of the world. It specifies the values of predicates or propositions that hold true at the beginning of the planning process. The initial state represents the situation from which the planner will start generating a plan.
In the example we can see that after the keyword *:init*, a series of state descriptions are defined. Here we initialize the total cost to zero so that we can start accumulating the cost metric. We also find the positions that are "empty" and would need to be filled with blocks and whether these positions are reachable (if they have a block or the ground underneath them). Concluding, we find the declaration of what blocks are in stock for the operator or for the robot arm.
The most essential part of the problem definition is the *goal*. The goal state defines the desired conditions or state that the planner aims to achieve, the target of the planning. It specifies the conditions or predicates that need to be satisfied for the problem to be considered successfully solved. The goal in this example is to find a solution for filling the position specified with a red cube, and we have two possibilities of action: the robot completes the task inserting the cube rc8 or the human completes it inserting rc5.

2.1.2 Planners and Solvers

After defining a Domain and a Problem, in order to obtain a solution plan a planner needs to be called. A planner is a computational tool or algorithm that is designed to generate a sequence of actions to solve a planning problem.
It takes as input a Domain and a Problem and returns ,where it is possible, an ordered plan that transforms the initial state into the desired goal state.
Planners in PDDL operate by exploring the space of possible actions and their effects to find a valid and/or optimal plan. They typically employ various search algorithms, heuristic functions, and planning techniques to efficiently navigate the search space and find a solution.



```
(define (problem complex)
 (:domain lego)
 (:objects
    robot_arm - robot
    operator - human
    p_02_10_4 p_05_13_3 p_02_11_3  p_02_10_3 platform - pos
    rc5 rc8 - red_cube
    yc1 - yellow_cube  )
 (:init
     (= (total-cost) 0)
     ( empty p_02_10_4 )
     ( under p_02_10_3 p_02_10_4 )
     ( in_stock rc5 operator )
     ( in_stock yc1 operator )
     ( in_stock rc8 robot_arm ) )
 (:goal
 (and   (or
                ( cube_at rc5 p_02_10_4 )
                ( cube_at rc8 p_02_10_4 )
     ))
) (:metric minimize (total-cost))
 )
```

Figure 2.3: Problem definition in PDDL

Overall, PDDL united with the state of the art planners ,empowers researchers and practitioners in the planning domain by providing a standardized language to describe and solve planning problems. Its adoption enhances problem-solving capabilities, facilitates knowledge sharing, and drives advancements in the field of automated planning and intelligent decision-making.

In this chapter, an evaluation will be conducted to assess the current state of the research in relation to this thesis's focus. In particular, the objective of this chapter is to identify the different types of planning techniques that are currently being experimented in correlation to HRC.
Planning is one of the most fundamental aspects of Human-Robot Collaboration. It is essential for coordinating actions between robots and humans, enabling a collaborative workflow. Without planning, the robot and human would not be able to perform actions in a synchronized manner, impeding effective collaboration.

2.1.3 Most used AI planning techniques: an overview

Task planning plays a crucial and established role in the fields of AI and robotics. Over the years, various AI planning methodologies have emerged and gained widespread usage. Additionally, recent advancements in the field of Machine Learning have led to the exploration and definition of new technologies in task planning.

This paragraph will introduce broadly some of the most used and relevant planning techniques and their main characteristics.

Planners derived from STRIPS

STRIPS (Stanford Research Institute Problem Solver) [12]planning language was developed in the late 1960s and it introduced a formal representation for the specification of planning problems that consist of a set of symbols, variables and actions and the solution consists in finding the set of actions that can transform an initial state into a set goal state, satisfying preconditions and effects associated to actions.
Planners derived from STRIPS typically employ search algorithms, such as depth-first search or A* search, to explore the space of possible actions and states. They make use of heuristics and search strategies to efficiently navigate the planning problem's state space and find an optimal or near-optimal solution. Some examples of STRIPS derived languages are:

- PDDL (Planning Domain Definition Language)[13]
- ADL (Action Description Language)[17]
- Golog (Goal Language)[18]
- O-Plan[19]

AI planning using machine learning techniques

In the last years machine learning techniques are increasingly being integrated into AI planning[20]. The idea is that by leveraging the power of machine learning algorithms and neural networks, planners can learn action models, heuristics and policies, enabling them to adapt and generalize to new environments.

Machine learning and specifically deep learning can be used do solve Ai planning problems[21] especially when these problems lie on big search spaces.

Some ways machine learning can help in Ai planning are:

- Action Models: Instead of manually specifying the action models (preconditions and effects) for planning, machine learning techniques can be used to learn these models from data.
- Heuristics: Heuristics play a crucial role in guiding the search process of a planner. Machine learning algorithms can learn effective heuristics from historical planning data or by training on a large number of planning instances.
- Learning from Experience: Reinforcement learning techniques can be employed to train planners through interactions with an environment. Planners can learn action selection policies by receiving feedback or rewards based on the success or failure of their plans. This iterative learning process enables planners to adapt and improve their strategies over time.

In general, very often a machine learning application might be over complicated for effectively solving smaller problems or problems that do not have the necessity of a learning curve.

Hierarchical task networks and AI planning

Hierarchical Task Network (HTN) is a planning methodology that focuses on decomposing complex tasks into more manageable subtasks. It is broadly used in AI planning[22] as it provides a hierarchical structure representing and organizing tasks, making it easier to find solutions and plan complex actions.

In HTN planning, then, a task is represented as a network of subtasks, where each subtask can be further decomposed into even smaller tasks until reaching primitive actions in the lowest level of granularity. This hierarchical decomposition allows for modular and structured planning.

Temporal and-or-graphs

Temporal and-Or-Graph (TOG) is a planning methodology in AI that combines both temporal reasoning and conditional branching to represent and solve planning problems. It extends the traditional And-Or-Graph (AOG) planning framework by incorporating temporal constraints and actions. It is rather useful in domains that involve temporal constraints and complex dependencies.

Markov decision process and its use in AI planning

Markov Decision Process (MDP) is a mathematical framework used in AI planning to model decision-making problems with uncertainty and sequential actions. It provides a formal framework for planning and decision-making under uncertainty. The main elements in a MDP are the **States** of the environment, that are reachable with **Actions** that can be taken with a certain probability defined by **Transition functions**. Some actions are more preferred or desired than others, and this relation is represented by **Reward functions**, a numerical value assigned to each state-action pair.

2.1.4 Relevant studies on task planning for Cobots

For the objective of this thesis, the evaluation and the study of the state of the art in this chapter will be focused on some main characteristics:

- the type of planning utilized in the study
- whether the planning approach is static or dynamic

- the consideration of constraints and difficulty associated with human actions
- the handling of these constraints in a dynamic manner

To be able to better grasp the classification made below, a brief explanation of the used criteria follows:

The **type of planning** simply describes the methodologies and techniques used to come up with the sequence of actions that define the collaborative task.
For **Static/Dynamic planning** it is intended that the planning has the ability to change during the execution of a task, so it means that the initial generated plan might be updated for every new unitary action realized or for every new information collected during the execution of the task. Then, a system that produces a different plan for different tasks but keeps the same plan through the whole task execution will be defined as static.
The **Difficulties/Constraints managing** criteria refers to the presence or absence of action cost consideration. This means that if a framework can change their planning to adapt to human constraints or if some fatigue parameters are considered when realizing the planning, then the criteria is respected. Such constraints can be a particular difficulty in completing a task, security risks or having one partner better suited for a specific job than another.
The **Validation Method** is the procedure used to evaluate the effectniveness of the proposed framework. For example, a framework can be evaluated through simulation but also through experimentation with an actual robot.

The authors of [23] propose a research on human robot collaboration on the disassembly of end of life products. Their focus is that of realizing a sequence planning that is able to take into consideration human fatigue and adapt the planning process to the human conditions. To do that, they assign some "class" to the actions. High complexity or uncertain actions are assigned to the human operator, that can handle them flexibly, while repetitive or heavy loading tasks will be taken by the robot. Doing so they consider human constraints in their planning. They decide that human fatigue is a function of time, for this, the planning approach is considered to be **static**. It is important to specify that the planning is not dynamic in the sense that it changes during the manipulation of the object, like in our experimentation, but that for each object a new plan is formulated in function of the time passed since the beginning of the work session. To obtain a plan they make use of the **Bees algorithm**, an heuristic algorithm that searches for an optimal solution between the feasible ones. The algorithm was experimented through simulation.

In the article [24] a new methodology called ProTamp is described and it proposes a probabilistic task and motion planning in the context of human robot

collaboration. The system is built to try to predict the human actions and thus the changes in the environment caused by human and robot. Also the deterministic environment changes are taken into account, wherever the robot can identify them. The probabilistic model determines the possibility of changes in the environmental state and tries to assign different tasks to the robot to avoid repetition of actions. The model used is based on **planning in Multi-layered action graph** built by recognizing the environment through sensors, and to predict the environmental changes a **Tree-diagram-based environment state graph** is built.
The planning described in the analyzed paper is considered **dynamic**, because it is able to change and re-plan during the execution of the task, reacting to the environment state changes, whenever they are detected. The study does **not take into account human constraints**, since it only makes use of the distance from the object and the elapsed time since starting to make its predictions.
The method was analyzed both through simulation and experimentation, and it was proved that the system allows for harmonious task execution.

The authors of [25] propose an interesting hierarchical framework for collaborative assembly planning. The model is composed of two layers or abstraction and allocation. To distiguish between robots and humans they use function costs, effectively **managing human constraints** in this way. Tasks are described through **hierarchical and concurrent hybrid state machines**, and **AND/OR graphs** and **A* graph-search** are used to produce the task descriptions. To the extent of the research description, the planning does not appear to be dynamic, it is instead **static** since it is not modified and reprogrammed automatically for each human decision and does not handle the human constraints dynamically. The system analyzed has been evaluated through experimentation.

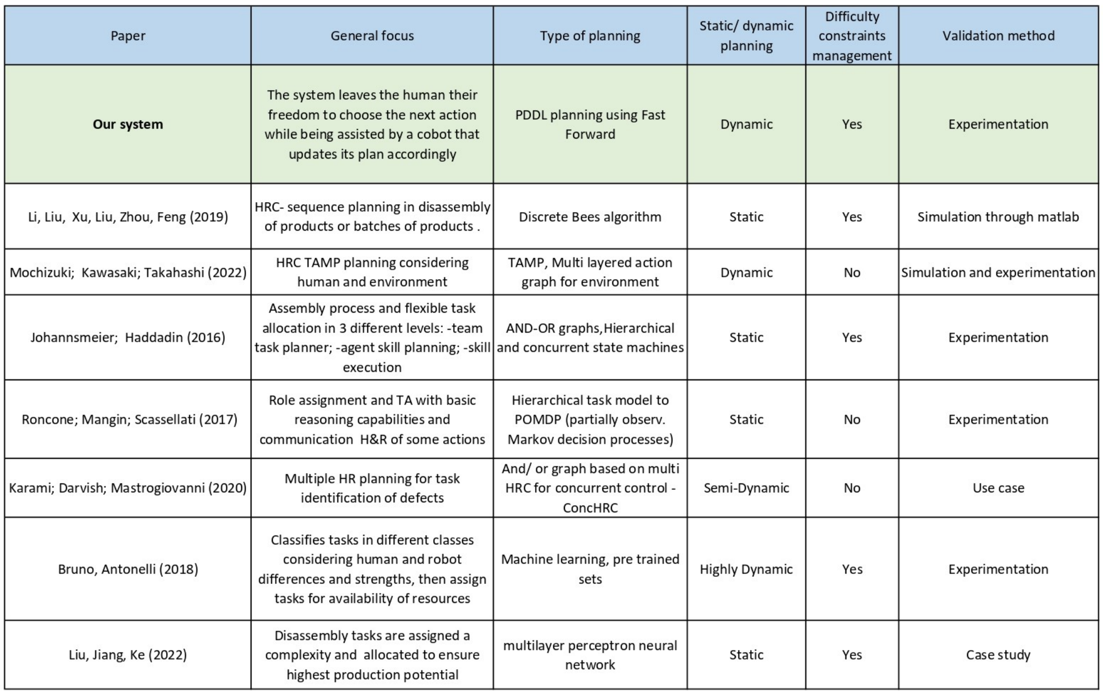
In [26] another task planner, this time based on **Partially observable Markov decision processes**, was proposed. The research is based on the idea that clear communication between robot and human is crucial to lower the mental load of the human and it removes the burden of the human in taking care of the robot. Communication is a crucial point also in our own research and, just as the authors of this cited paper, we believe it is essential for fluid collaboration. In this analyzed paper, the human needs to communicate his actions which can still be burdening for the operator and time consuming.
In their experiment, the roles of human and robot are different and explicitly decided beforehand. The planning, for that, is **not dynamic** as there is no need to avoid choosing colliding actions in this scenario. For the same reason, the planner does not take into account the cost of actions or human constraints.

The paper [27] describes a new methodology that allows humans and multiple heterogeneous robots to interact between them to perform a defect inspection task. In this study the tasks are unequivocally assigned to the different human or robot operators which makes the task quite static from the point of view of our own research. Even though the human and robots actions are predefined, we still find **some dynamicity** in the choice of the robot's next action, since it is decided by the human output of his previous action. The task planner is then called once again and the robot actions are updated accordingly. **And/or graphs** are used for task allocation in this model.

The authors of [28] propose a system that makes use of dynamic task classification and then assignment of the task to different operators, human or multiple different robots. The study is quite interesting since it first analyses the type of task to discover which one is the most suited operator to assign the action to. A plan is then built upon the classification.
The cited research is considered to be **highly dynamic** since at first the dynamism is found the assignment of a task to a specific class, after that the tasks are assigned considering available resources and periodically reassigned to match the change in resource usage.
Furthermore, the **human and robot constraints** are considered since the tasks are classified into four categories: only executable by human, only executable by robot, executable by both H and R and only executable in collaboration. This study makes use of **machine learning techniques** and pre-trained sets for the task classification.

Similarly, a Task allocation method is proposed by [29] for the disassembly of automotive traction batteries in HRC. This method is based on the complexity analysis of the disassembly taking into consideration human and robot **constraints** and qualities. After the evaluation of the complexity of the task, the action allocation happens through the use of a **multilayer perceptron neural network**. The **task assignment and order is static** since it is not changed after the task allocation process. The strong suit and our interest on this research is their focus on the complexity and constraints analysis.

Overall, the studies reviewed above generally lack the integration of dynamism and consideration of human constraints.
Additionally, while PDDL has demonstrated to be effective in numerous task planning and robotics studies, its application in combination with dynamism and constraint consideration appears to be limited.
In contrast, our research aims to address this gap by proposing a **highly dynamic** and **cost-conscious** system that is **environment-aware**, leveraging the capabilities of a camera to observe human and robot actions. Notably, our approach



Paper	General focus	Type of planning	Static/ dynamic planning	Difficulty constraints management	Validation method
Our system	The system leaves the human their freedom to choose the next action while being assisted by a cobot that updates its plan accordingly	PDDL planning using Fast Forward	Dynamic	Yes	Experimentation
Li, Liu, Xu, Liu, Zhou, Feng (2019)	HRC- sequence planning in disassembly of products or batches of products .	Discrete Bees algorithm	Static	Yes	Simulation through matlab
Mochizuki; Kawasaki; Takahashi (2022)	HRC TAMP planning considering human and environment	TAMP, Multi layered action graph for environment	Dynamic	No	Simulation and experimentation
Johannsmeier; Haddadin (2016)	Assembly process and flexible task allocation in 3 different levels: -team task planner; -agent skill planning; -skill execution	AND-OR graphs,Hierarchical and concurrent state machines	Static	Yes	Experimentation
Roncone; Mangin; Scassellati (2017)	Role assignment and TA with basic reasoning capabilities and communication H&R of some actions	Hierarchical task model to POMDP (partially observ. Markov decision processes)	Static	No	Experimentation
Karami; Darvish; Mastrogiovanni (2020)	Multiple HR planning for task identification of defects	And/ or graph based on multi HRC for concurrent control - ConcHRC	Semi-Dynamic	No	Use case
Bruno, Antonelli (2018)	Classifies tasks in different classes considering human and robot differences and strengths, then assign tasks for availability of resources	Machine learning, pre trained sets	Highly Dynamic	Yes	Experimentation
Liu, Jiang, Ke (2022)	Disassembly tasks are assigned a complexity and allocated to ensure highest production potential	multilayer perceptron neural network	Static	Yes	Case study

Figure 2.4: Literature review recap

eliminates the need for explicit communication of decisions from the human, offering a more seamless and intuitive collaboration between humans and robots.

Chapter 3

Description of the Framework

As mentioned in the previous chapters, the main issue tackled by this research is how to make a cobotic system effective by considering human variabilities and constraints in its planning.
This gives rise to two main key challenges:

- Allowing humans the freedom to act as they choose while enabling the robot to adapt accordingly,
- Ensuring that the robot is aware of human constraints and plan tasks in a way that reduces mental effort and ensures safety.

Furthermore, it is important for the human-robot collaboration to be seamless and user-friendly, so that users find it easy to learn and work with the system.

The proposed solution is based on the extension of the work made on the Cobotic system framework realized by the Marvin team- LIG lab Grenoble in France, that I joined in my master research. The modifications and contributions to the system will try to tackle the problem at hand.

In the implementation of the framework, we choose to use **PDDL** (Planning Domain Definition Language) for task planning, which enables us to generate efficient and optimized plans for the collaboration between human and robot.
To facilitate seamless communication, we incorporate a user-friendly **Graphical User Interface (GUI)** that serves as a medium for interaction between the human and the robot. Additionally, we integrate a **vision system** that automatically updates the system's state, allowing for real-time awareness of the task progress.
In our system, both the human and the robot know what the task goal is.

The human has the freedom to decide when they need the robot's help, and to make things clear, the robot openly shares its planned actions through the GUI, ensuring clear understanding and coordination between human and robot throughout the collaborative process.

In the next section we will explore the details of the proposed implementation.

3.1 The Cobotic System

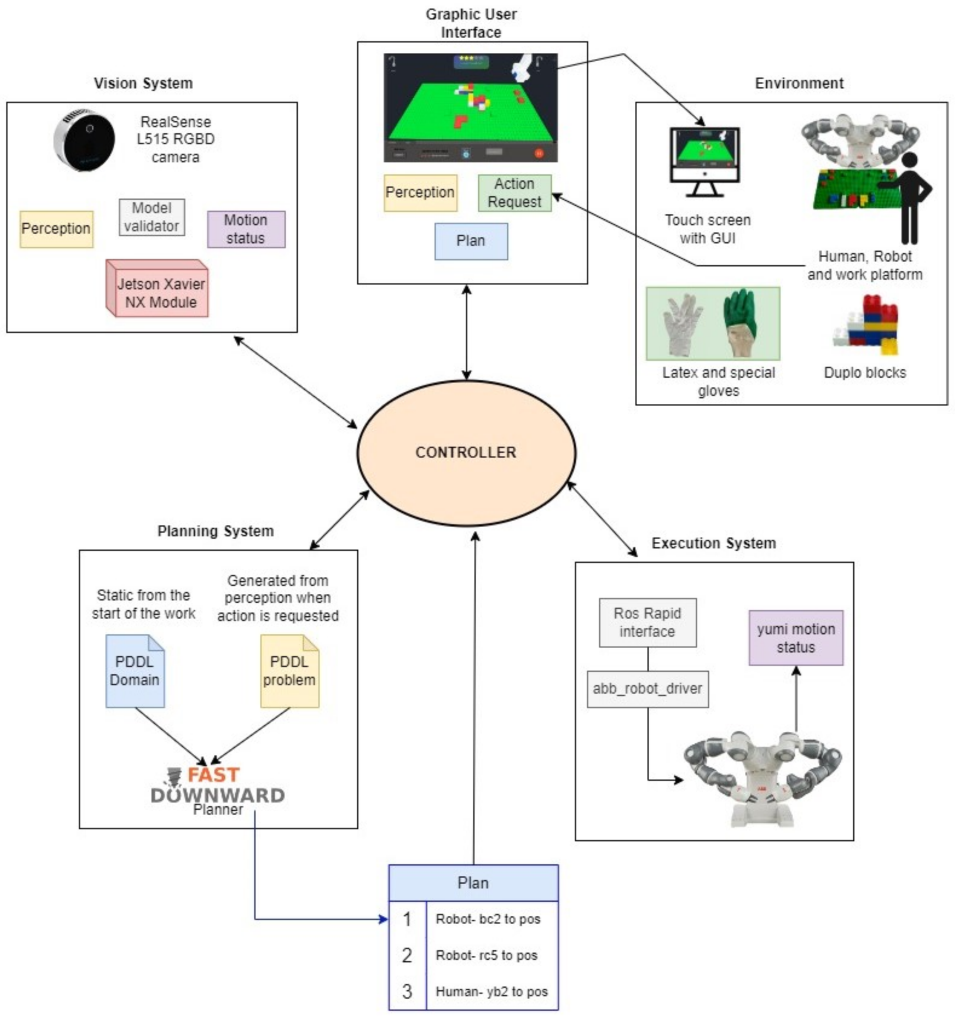


Figure 3.1: Illustration of the framework

The proposed system utilizes different techniques and for this, separate modules and environments are necessary, as well as a way for them to communicate between each other and exchange information with the real world.

3.1.1 Environment description

The environment of the system is what we can see physically in the real world. The work setup is composed of a **green lego work platform** placed on a plexiglass table, a **stock of lego duplo blocks** of various colors, two different types of **gloves** (latex and special security gloves) for the manipulation of different colors of blocks.

The actors of the system are **the human and the robot**, which are also considered to be an important part of the system environment.
The robot in use in our system is the YUMI dual arm industrial robot from ABB.

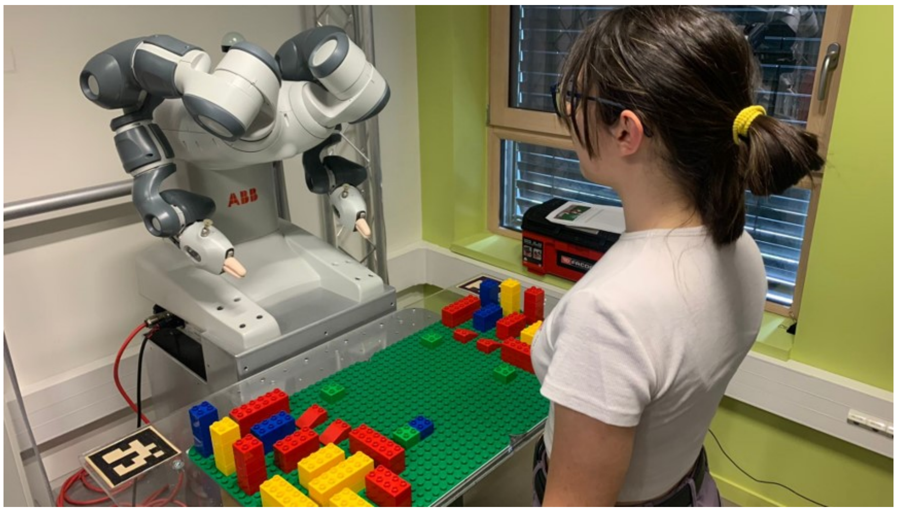


Figure 3.2: Collaborative environment: Human-robot setup

It will be described throughly in the next sections.
Another essential part in the environment setup is the Camera RealSense L515 RGBD , that is the main communication instrument for the Vision System with the real world.
The most important link of communication between the cobotic system and the human is the **Graphic User Interface**. Through the GUI accessible from a Touch Screen the human is able to communicate to the robot when he wants the robot to act, and to receive important information about the state of the task and the robot status and perception.

3.1.2 The Graphic User Interface

The Graphic User Interface represents the main channel of communication between the human and the system. It is accessible through a touch screen placed in the vicinity of the human location.

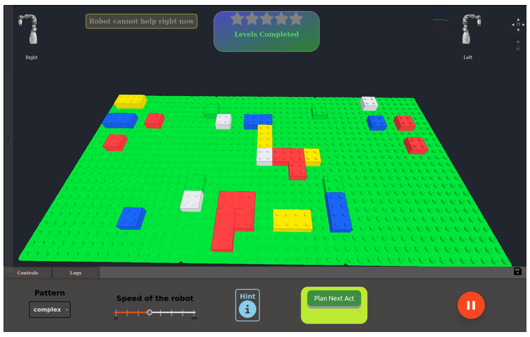


Figure 3.3: GUI - task at level 1 and "robot cannot help " message

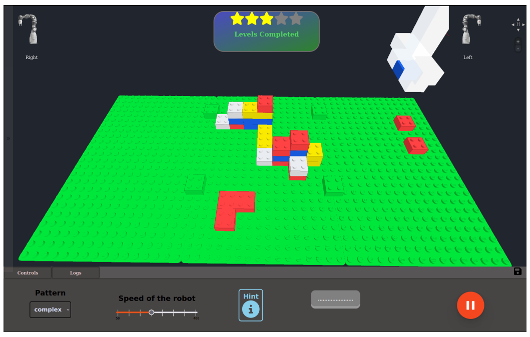


Figure 3.4: GUI - task at level 3 and robot communicating its next action

The GUI presents in the main screen a reproduction of what is the perception of the task that the camera is seeing and that it is elaborated by the **Controller**. The representation of the task on the green table is needed:

- To communicate to the human if they make errors, for example, a red X appears on top of a misplaced block
- To show the human what the robot is perceiving and being able to check if there are errors in the robot perception due to vision or environment problems
- For the robot to be able to show the human its next planned action, as we can see in Figure 3.4. Before an action is physically performed by the robot, a robot arm appears in the GUI to pick up the selected block to put it in the chosen position.

The last point is essential so that the human, once he is informed of the robot's next action, can choose to continue to work alongside the robot without interfering with the robot's work.

As we can see from the Figure 3.3, the GUI is equipped with an interactive bar at the bottom of the screen where the human can interact with the system.
The human needs to choose the type of pattern to work on, and he can choose to increase or decrease the speed of movement of the robot.

Additionally, the human can ask for Hints on their next action when needed or wanted.

Successively, to ask the robot to perform its next action, the green button "Plan Next Act" needs to be pressed. In this way the human has full control on when the robot should help him and he can count on the fact that he will not be surprised by some robot actions. In this way, the human will interact with the screen and will be able to see and launch the robot action (and know how to avoid it). Moreover the pressing of the "Plan Next Act" button ensures that the replanning is done when the human hand is outside the workplace in order to have a valid perception information to guide the planning, since the screen is placed far from the working table.
We implemented this security also to make the system more acceptable by individuals who have never been in contact with an industrial robot before and could experience feelings of mistrust or fear.
Finally, the bar is equipped with a start/stop button to be pressed at the beginning of the collaborative task when the human is ready to start, and in case of emergencies to stop the task process.

In the top part of the screen we find some output communication messages: in figure 3.3 the robot is communicating that it cannot help right now.
At the top center we find a display of "stars" that represent the current level the task is at. The task we experimented with had 5 levels corresponding to different floors of work.
The human and the robot will work on the task level by level, and only when a new star appears in the screen is the level considered completed. The stars work both as a reward for the human and as a way to know what level is being completed at the moment. When the task is finished, and all 5 stars are collected, a congratulating message appears at the center of the screen letting the human know the task was successful as in figure 3.5.

3.1.3 The Vision System

The vision system main link to the real world is the RealSense L515 Camera that is positioned above of the workstation to be able to see the lego blocks in the workspace. The camera is connected to a Jetson Xavier NX where the code for the vision module is stored and ran, and where the first analysis of the perception happens.
The main modules that run in the vision system: the vision launch, the vision logic and the model validator.
The vision launch is needed to activate the camera and correctly identify the

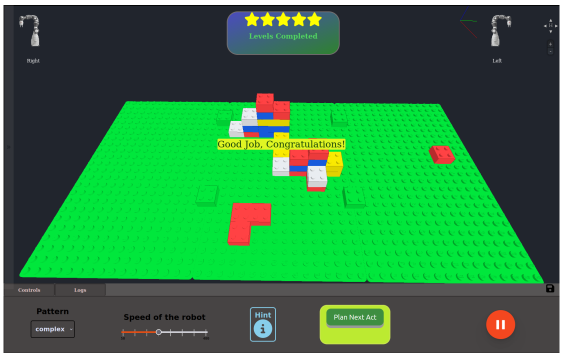


Figure 3.5: GUI- task completed, congratulations message displayed

different types of blocks and their positions. It is the main package used for Detection.
The vision logic is a Geometric reasoning module to infer from the vision some information on the status of the legos, for example which Lego block was picked and then placed, amd the new position of the block.

The model validator is needed to validate that the pattern assembled until now is correct, to update the level status when a level is terminated, to manage the pick and place of cubes from the point of view of the camera (the robot arms are going to cover the camera vision). The motion status received from the Execution System is used for this scope.

The model validator is able to know the ground truth of the task because it is contained in preloaded JSON databases. For example, we have a file containing the initial stock of blocks for both human and robot and another file for the finished task.
We report here a snippet of the json database for the completed task:

```
{
  "complex": {
    "p_01_08": {},
    "p_02_08": { "1": "w" },
    "p_02_09": { "1": "r", "2": "l", "3": "w" },
    "p_02_10": { "1": "b", "2": "b", "3": "y", "4": "b", "5": "r" },
    "p_02_11": { "1": "b", "2": "b", "3": "y", "4": "o" },
    "p_02_12": {},
    "p_02_13": {},
    "p_02_14": {},
 ...
```

The correct solutions for some positions are reported here. As we can see, for every position we have the color of block that should be placed there and at which level. If the vision system detects a block in a position that is not listed in the database of the completed task, the model validator will detect it and tell the GUI to show the error to the human. The block will only be considered corrected if it is placed in one of the listed positions for the specific block color and type.

To easily translate the perceived workspace to a manipulable entity, each block has been assigned a code of the type "$L\ t\ n°$ " where L= Letter corresponding to a color; t= type of block (c for cubes and b for bricks) ; $n°$= a cardinal number. For example the code "bc4" corresponds to a blue cube indexed 4.
Similarly, each position in the workspace will corresponds to a cartesian x-y-z axis position p of the form *"p_xx_yy_zz"*

The pattern registered up to this moment is converted using block codes and positions in a similar way as it was shown in the json code above.
All the data collected by the vision system is then sent to the State Controller where some modifications are made to the perception state received and then the perception is saved and sent to the GUI to be displayed.
When the human asks for the robot's next action through the GUI, a trigger is sent to the controller and the last saved perception state is sent to the Planning system so that the planner can work with the last updated state of the task.

3.1.4 The Execution System

The execution system of the framework is mainly represented by YuMI Dual-arm IRB 14000 collaborative robot and the necessary software needed to allow its controlled movements.
Yumi is a collaborative robot developed by ABB. Yumi stands for "You and Me",

Figure 3.6: Yumi dual arm collaborative robot

meaning humans and robots working together in collaboration. Some of its main characteristics :

- Design and Features: Yumi is designed as a dual-arm robot with human-like flexibility and dexterity, with 14 axis of movement freedom. It has compact dimensions, lightweight construction, and soft padding to ensure safe interaction with humans. The robot has a built-in controller and a wide range of motion, allowing it to perform intricate tasks with high precision.
- Collaborative Capabilities: It is specifically designed for collaborative applications, working alongside humans in various industrial settings. It is equipped with advanced safety features, including sensitive joint torque sensors and collision detection technology. These features enable Yumi to safely operate in close proximity to humans without the need for safety barriers.
- Easy Programming: Yumi is designed to be user-friendly and easy to program. It uses ABB's intuitive programming interface, allowing both experienced and novice users to quickly teach the robot new tasks. Programming can be done through manual guiding or offline programming, depending on the necessities.
- Wide range of applications: Yumi is suitable for a wide range of collaborative tasks in industries such as electronics, automotive, and consumer goods. It can perform assembly, pick-and-place operations, packaging, machine tending, and other tasks that require precision and flexibility. Yumi's dual-arm configuration enables it to perform tasks that would typically require two separate robots or a human and a robot working together.

Yumi is connected to our framework through the use of a personalized ROS interface. The actions that need to be performed by the robot are sent from the controller to the robot interface. Similarly, the Yumi motion status is collected from the robot interface and sent to the controller to be used as necessary in other parts of the system.

3.1.5 The Planning System

The planning system plays a prominent role in enabling the dynamic nature of the framework and requires careful implementation.
The planning system is designed to be dynamic, generating updated plans whenever the human operator requests the robot to perform an action. This involves updating the problem based on the current status of the vision system.
There are two essential scripts involved in generating a plan: the PDDL (Planning Domain Definition Language) domain and the PDDL problem. The PDDL domain is created once and remains static, while the problem is dynamic and changes

with each request. This is because the base environment the task is conducted on and the definition of the actions does not change while the problem analyzed (type, number and position of the blocks) is the one that changes. The necessity for dynamism of the pddl problem will be explained in greater detail in the next section.
Once the new PDDL problem is generated, both scripts are sent to the planner for solving and generating an appropriate plan.
The planner chosen for the system is the Fast Downward planner.

Fast Downward

Fast Downward [30] is a classical planning system that uses heuristic search to solve general deterministic planning problems. In the last years, it has achieved success in planning competitions and performs well on benchmarks, providing insights into search enhancements. It is considered the state of the art for PDDL based planners and it is known for its efficiency and its ability to handle large-scale planning domains.
In its last releases, Fast Downward also supports the specification of action costs and disjunctive-preconditions to find an optimal solution with respect to a specified cost metric.

After the generation, the plan is sent to the controller to be reordered and corrected before calling the robot to action.

3.1.6 The Controller

The state controller is an automaton that serves as the central link connecting all the modules within the described framework. It acts as a vital intermediary for transmitting information between different parts of the system.
The most important links that the controller allows are:

- Facilitating communication between the vision system and the planner. This enables the planner to utilize the most up-to-date work state for generating the PDDL problem file. Additionally, the GUI also receives the updated perception from the controller and can constantly display an updated perception of the work for human observation.
- Enabling ordered information exchange between the planner and the system. This exchange is crucial for organizing the next action of the robot and for displaying the selected action in the GUI.
- Ensuring seamless coordination between the requests made by the human through the GUI. When an action request is made in the GUI, the trigger is

sent the controller that initiates the process of the next action by calling the planner to generate the plan and initiate the robot's subsequent action.

Overall, the controller serves as a central component in controlling the flow of information between different parts of the system. It guarantees that the exchanged data is continuously updated and correctly processed, supporting the smooth operation of all processes within the framework.

3.1.7 The functioning of the framework

With these implementations to the system, we can foresee how the system is going to work when a human and a robot need to collaborate to realize a common predefined task.

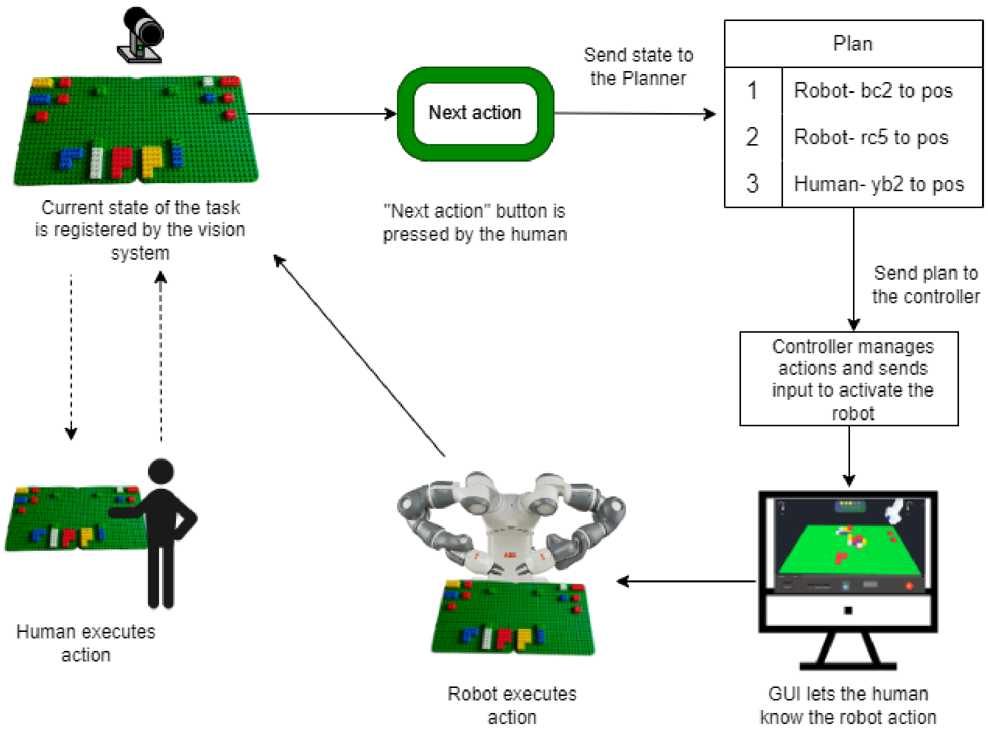

Current state of the task is registered by the vision system
Next action
"Next action" button is pressed by the human
Send state to the Planner
Plan
1 Robot- bc2 to pos
2 Robot- rc5 to pos
3 Human- yb2 to pos
Send plan to the controller
Controller manages actions and sends input to activate the robot
GUI lets the human know the robot action
Robot executes action
Human executes action

Figure 3.7: Execution of the system

At the beginning, the state of the working table is identified by the perception system and it is always updated for every new action realized from Robot or Human.
At that point the human can either choose to perform a manipulation action of his/her choice or to ask the robot to perform the next action. When the robot is asked to perform an action, the planner is activated and by reading the current

state of the task reached, it is going to realize a plan that gives precedence to the robot actions and if needed takes into account human constraints.
The generated plan is a list of actions that indicate the Actor, the block to be manipulated and the new position to place the block in.
Some actions for the human will be generated if there is no possibility for the robot to help any further (for example if it is missing the necessary blocks in its stock). In that case the GUI will display a message for the human, letting him know that the robot cannot help any further for now. At that point the human has the possibility to either directly do the required action or ask for an Hint from the system if he does not know how to proceed. After every action the vision system will update the state of the task and the human can check the correctness of the system. The cycle of actions by both partners will continue until the task is completed. When the task is terminated, the GUI will project a message communicating the success of the job.

3.2 The decision making PDDL system

The decision-making or planning system plays a critical role in generating action plans for the robot, while also considering potential constraints and the cost of human actions. It can be likened to the brain of the robot, constantly assessing and determining the best course of action.

However, the planner's objective extends beyond dynamic planning. It also aims to generate efficient plans that account for any limitations or preferences the human may have in their actions.

To achieve this, we have implemented a static Domain capable of considering action costs through various actions. This Domain is created once at the beginning and remains consistent throughout the system.
In our implementation, we have developed two distinct PDDL domains, each accounting for actions with different costs.
These domains are selected based on the specific requirements of the task.

3.2.1 The PDDL Domain

In this paragraph we will illustrate some important parts of the PDDL domain that will be used in the experiments when the robot is asked to consider human constraints such as that their difficulty in managing red cubes and bricks (more information will be given in the Experiment chapter).

The declaration of the domain is realized as in figure 2.1 and the definition of the types is shown here:

domain

```
    (:types
       red_cube - cubeD
       blue_cube - cubeB   yellow_cube - cubeB  white_cube - cubeB
       red_brick - brickD
       blue_brick - brickB   yellow_brick - brickB
       cubeB - cube   cubeD - cube
       brickD - brick   brickB - brick
       cube - lego   brick - lego
       lego - object
       robot - agent   human - agent
       agent - object   pos - object
    )

```

From the code above we can see how the red cube and the red brick definition is different from the definition of all other blocks (cubeB and brickB, to mean cube/brick Basic), as they were assigned to be of the type cubeD and brickD (meaning cube/brick dangerous). This difference in definition will be useful later on when defining the actions of the domain that respect the different costs.
The types cubeB and cubeD are then recursively assigned to type cube, that is assigned to type lego and then lego to object. A similar definition is made for bricks.
Robot and human are defined as agents that correspond recursively to objects.
The predicates are declared as in figure 2.1 and they indicate:

in_stock :	if a certain lego is in the agent stock or not
cube_at & brick_at :	what is the position of a certain cube or brick
empty :	if a specified position is empty or not
under :	to indicate that position2 is under position1

The next step of our domain definition is the declaration of a function:

```
    (:functions
        (total-cost) - number )
        )
```

that is necessary to declare the total cost variable that will be used as an accumulator by the problem.

The last essential part of the Domain is the action definitions. An example of an action is shown here:

domain

```
        (:action pick_place_cube_robot
    :parameters (?r - robot ?c - cubeB ?p ?p0 - pos)
    :precondition
        (and
            (in_stock ?c  ?r)
            (empty ?p)
            (under ?p0 ?p)
            (not (empty ?p0))
        )
    :effect
        (and
            (not (in_stock ?c  ?r))
            (cube_at  ?c ?p)
            (not (empty ?p))
            (increase (total-cost) 1)
        )
    )

```

This action corresponds to the robot action of picking and placing a basic cube. The parameters indicated are: the robot, a specific basic cube and two positions, the first where the cube currently is and the second where the cube needs to be placed.
The preconditions indicated right after are essential because if they are not satisfied the action cannot happen. In this case it is required that the cube is in the robot stock, the arrival position is empty, the positions both have something underneath and finally that the starting position is not empty (it needs to still have the cube to be picked up).
The effects that happen after the action is executed are that the cube is not in the robot stock anymore, the cube is placed at the new position so that it is not empty anymore, and finally that the pick and place of this specific cube causes an increase on the total-cost of the plan that is equal to 1.

The declared actions are:

:action pick_place_cube_human & :action pick_place_brick_human : human action of picking and placing a normal cube or a normal brick. The increase in **total-cost is 2**

:action pick_place_cube_robot & :action pick_place_brick_robot : robot action of picking and placing a normal cube or a normal brick. The increase in **total-cost**

is 1

:action pick_place_cube_human_dangerous & :action pick_place_brick_human_dangerous : human action of picking and placing a dangerous cube or a dangerous brick. The increase in **total-cost is 20**

:action pick_place_cube_robot_dangerous & :action pick_place_brick_robot_dangerous : robot action of picking and placing a dangerous cube or a dangerous brick. The increase in **total-cost is 0**

The differences in increases of the total-cost are necessary to ensure different priorities for the scheduling of the robot actions. With these values, a human action for a dangerous block will only be scheduled if strictly necessary, and the priority of scheduling will be given to the robot action of handling the dangerous blocks, and then subsequently to the robot actions for normal blocks and when necessary the human actions for normal blocks.
With these configurations, we are able to guarantee that the planner will choose to plan the robot actions with more priority and the human actions only when convenient or strictly necessary.

3.2.2 The generation of the PDDL problem

Since the problem needs to change every time a new plan is generated and every time an action is requested by the human operator, it needs a dynamic way of creating a problem starting from:

- the perception of the vision system
- the remaining blocks in the human and robot stack
- the solution of the task for the specific level considered

The proposed solution uses a python script which, by taking as input the data mentioned above, returns a problem fitting the current necessities.

The problem generator works as follows:

Define the problem name and domain First it writes on the PDDL file the initial definition of a problem in the form problem

```
    (define (problem complex)
       (:domain lego)
```

Define the objects It appends the list of objects, such as in the following example

problem

```

(:objects
     operator - human
     robot_arm - robot
     p_05_13_3 p_02_11_3 p_02_10_3 p_02_09_2 p_05_13_2
              p_05_12_2 p_02_10_2 p_02_11_2 platform - pos
     wc2 wc3 wc4 - white_cube
     rc8 rc5 rc6 rc7 - red_cube
     yc1 - yellow_cube
     bc2 bc3 bc4 - blue_cube
     yb2 yb3 - yellow_brick
     bb2 - blue_brick
     rb2 - red_brick
 )

```

The list of objects always contains the two actors (human operator and robot arm), the list of positions corresponding to the blocks and the list of the blocks with their corresponding code taken from the remaining (human or robot) stock list.

Define the initialization The initialization of the problem is necessary to estabilish the conditions of the space of work. An example of such initialization is:

problem

```
    (:init
     (= (total-cost) 0)
     ( in_stock wc2 operator )
     ( in_stock wc3 operator )
     ( in_stock wc4 operator )
     ( empty p_05_13_3 )
     ( under p_05_13_2 p_05_13_3 )
     ( under p_02_11_2 p_02_11_3 )
     ( in_stock rc8 robot_arm )
              ...
              ...
 )
```

The initialization of the total cost is useful for the accumulation of the costs of actions so that the planner can choose the best actions to insert in the plan.
The initialization of the legos in stock and the positions empty or under is necessary for respecting the preconditions.

Define the goal The definition of the goal is created using a generated list of positions and legos to be inserted to complete the level. From this list, a structure made of and & or is formed.

problem

```
:goal
        (and (or
                ( cube_at rc8 p_05_13_3 )
                ( cube_at rc5 p_05_13_3 )
                ( cube_at rc6 p_05_13_3 )
                ( cube_at rc7 p_05_13_3 )
        ) (or
                ( cube_at bc2 p_02_10_3 )
                ( cube_at bc3 p_02_10_3 )
                ( cube_at bc4 p_02_10_3 )
    ) ) )
```

In this explicative example, the position p_05_13_3 needs to be filled with a red cube and the position p_02_10_3 needs a blue cube.
The problem generator uses the list of cubes in stock to generate the different action possibilities for each action necessity. For example, since we have four different red cubes that can fulfill the necessity of position p_05_13_3, then they are all listed as possible actions. Then the planner will choose the best action between them considering the cost.

Define the metric To obtain an ideal plan, the total-cost needs to be as small as possible. We indicate this need through the last line of the generated plan:

```
    (:metric minimize (total-cost))
```

so that the total-cost can be minimized by the planner, when possible.

3.2.3 The Fast Forward Planner to generate the solution

The fast forward planner is called and it starts by analyzing the initial state of the problem and constructing an abstract representation of the domain. It then uses this representation to generate a search graph, where each node corresponds to a different state of the problem. The search algorithm systematically explores the graph, evaluating the cost of each potential solution path based on the specified cost metric.

To guide the search process, Fast Downward employs heuristics that estimate the cost of reaching the goal state from each explored node. These heuristics provide an informed estimate of the remaining cost, which helps the planner prioritize the exploration of more promising paths.

Fast Downward uses a best-first search strategy, which means it prioritizes expanding nodes with lower estimated costs. This allows the planner to focus its efforts on the most promising solution paths and potentially reach an optimal solution more efficiently.

Throughout the search process, Fast Downward keeps track of the best solution found so far, gradually refining it as it explores the search space. Once a goal state is reached, the planner returns the optimal solution path, taking into account the cost metric specified.

It is important to specify that the optimal solution is not always the less costly, but also the one that took less time to find while still optimizing following the cost metric.

After having generated the plan, the solution is sent to the controller where some adjustments are made, such as reordering the actions in a necessary order or canceling actions that became unfeasible in the meantime.

Chapter 4

The Experimental Scenario

The creation of the experiment is an essential part of this thesis's contribution and it is needed to evaluate the performances of the framework in a real environment and in contact with humans. The experiment will help us gather the needed feedback from real people's experiences that will help us evaluate the system more critically.

The idea behind it is that of creating an experimental design that is able to answer some specific questions on the capabilities of our framework.

To guarantee the validity of the methods besides the technical aspects, the experimental design and the experiments were created in collaboration with a work psychology and ergonomics expert, in the context of a PHD study with a focus on the effects of Human robot collaboration.

The **hypotheses** we set for this experiment and that we want to explore are:

1. The adaptive condition of the cobotic system will have an effect on the participant's performance (completion time, errors, gestures), on the risks taken (number of dangerous blocks handled) and on the delay of usage of the special glove.
2. The adaptive cobotic system will result in reduced perceived workload and perceived acceptability(usability perceived, coherence perceived, pleasure perceived, usefulness perceived, social influence and trust), for the human participants.
3. The operator's expertise (expert vs non-expert) will have no effect on the acceptability (usability perceived, coherence perceived, pleasure perceived, usefulness perceived, social influence and trust). The operator's expertise will have an effect on the workload, on the performance (number of errors, time

and number of gestures), on the risk taken and on the security constraints of the cobotic system.

The main idea of the experiment is that of **adding a constraint** for the human that the robot will supposedly help with, and observing how **human variabilities** will be managed by the system.

We simulate a **chemical work environment** where the human and robot need to handle some dangerous or less dangerous chemical substances, represented by different color duplo blocks.

The duplo blocks are used to simulate substances because of their regular shapes and intuitiveness for manipulation both from humans and robots. The main objective of the experiment is the evaluation of the decisions made by the robot and the duplo blocks are designed to be assembled and stacked one on top of the other and their colors are designed to be very differentiable, which makes them a perfect item for the basic type of manipulation of objects that we want to obtain from this experiment.

The substances will have **different levels of danger** that will be represented by red blocks for the most dangerous substance and all the other blocks will be regular substances.

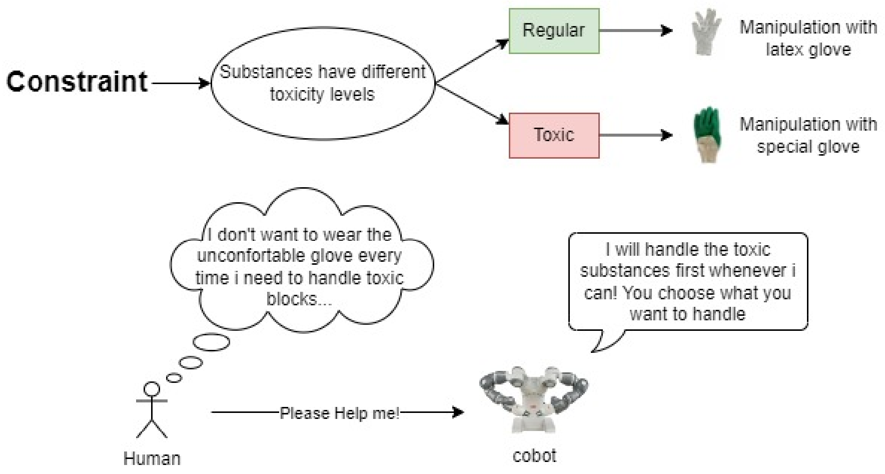

Constraint
Substances have different toxicity levels
Regular
Toxic
Manipulation with latex glove
Manipulation with special glove
I don't want to wear the unconfortable glove every time i need to handle toxic blocks...
I will handle the toxic substances first whenever i can! You choose what you want to handle
Please Help me!
Human
cobot

Figure 4.1: Illustration of the constraint

To be able to appreciate the **usefulness of the cobotic system in handling the constraint**, half of the participants will participate on the collaborative task while the cobotic system does not know about any constraints (it will not give precedence or more importance to the manipulation of the toxic substance) while the other half of the participants will be helped by the cobotic system that will know about the constraint and will prioritize the manipulation of toxic substances in its planning. In all cases all the participants will have to follow the constraint and wear the gloves.

The human and the robot will be given a task to complete that is based on the assembly of a pattern with the use of lego blocks. Both partners will be aware of the task, but the human will be able to see it only through a picture, which can be limiting as it will be explained later on.
For the manipulation of the legos the human will be asked to wear different gloves depending on the toxicity level of the blocks, such as in real scenarios where a human is required to wear special protective gloves when handling toxic substances. Moreover, we thought it would be significant to add a secondary variability besides the cognitive one that every human has: **The expertise level**. Half of the human

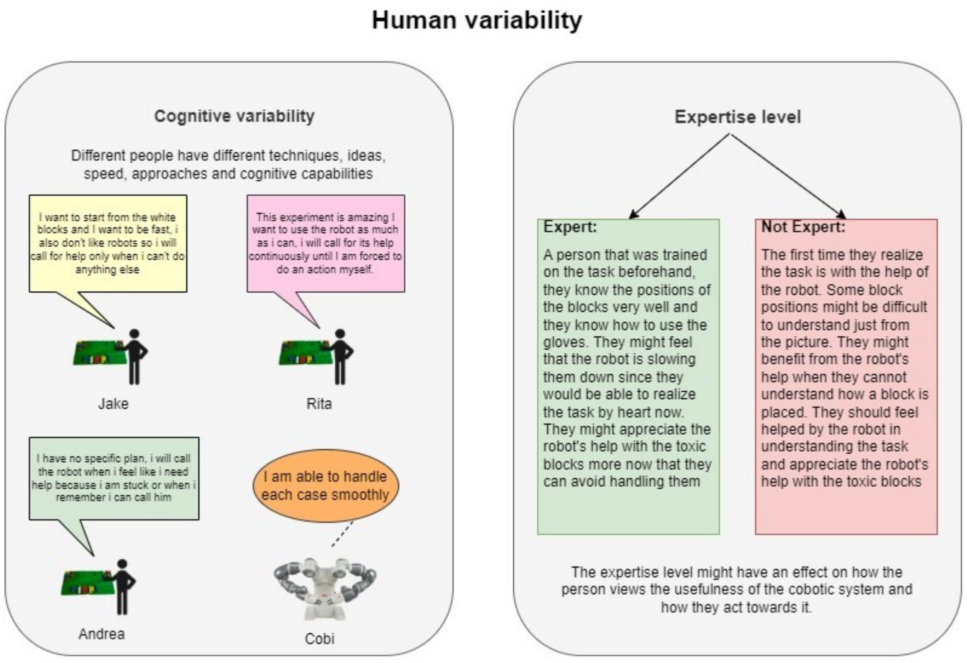

Human variability
Cognitive variability
Different people have different techniques, ideas, speed, approaches and cognitive capabilities
I want to start from the white blocks and I want to be fast, i also don't like robots so i will call for help only when i can't do anything else
This experiment is amazing I want to use the robot as much as i can, i will call for its help continuously until I am forced to do an action myself.
Jake
Rita
I have no specific plan, i will call the robot when i feel like i need help because i am stuck or when i remember i can call him
I am able to handle each case smoothly
Andrea
Cobi
Expertise level
Expert:
A person that was trained on the task beforehand, they know the positions of the blocks very well and they know how to use the gloves. They might feel that the robot is slowing them down since they would be able to realize the task by heart now. They might appreciate the robot's help with the toxic blocks more now that they can avoid handling them
Not Expert:
The first time they realize the task is with the help of the robot. Some block positions might be difficult to understand just from the picture. They might benefit from the robot's help when they cannot understand how a block is placed. They should feel helped by the robot in understanding the task and appreciate the robot's help with the toxic blocks
The expertise level might have an effect on how the person views the usefulness of the cobotic system and how they act towards it.

Figure 4.2: Illustration of the variabilities

operators were trained on the completion of the task alone before working on it in collaboration with the robot. This class of participants will be called The Experts.

4.1 Details and specifications

Indicators of performance:

- Number of errors: piece placed and replace elsewhere
- Completion time: the completion time of the model
- Number of red blocks handled -> Number of times the special glove was used
- Task level when the special glove was used for the first time

Indicators of workload:

- Results of NASA TXL (Hart, 2006; Hart & Staveland 1998; Rubio al., 2019; Webb al., 2021)
- Number of gestures

The gestures are all movements of the upper limbs within the work zone observed between the start of the experimentand the end.

Indicator of acceptability:

Acceptability auto-reported: results of the questionnaire (Venkatesh & Bala, 2008).

Participants

Participants profile The participants are adults, without any color vision impairment and with no pain in the upper limbs prior to the experiment.

Why ? These participants are required to work on an assembly task with the cobot. The goal is to observe them during the task to be able to compare between two conditions. They need to be pain free in the upper limb to realize the task. Since they will have to construct a model of colored Duplo, they must not have any color vision impairment.

Number of participants

The conditions "expertise" and "adaptation to constraints" are inter-participants. The ideal number of participants is 40, so that there are 20 participants in each of the two conditions so that there are enough cases to confirm statistical results. We were able to carry out 40 experiments, of which 38 succeeded and 2 were canceled for unexpected technical problems with the robot.

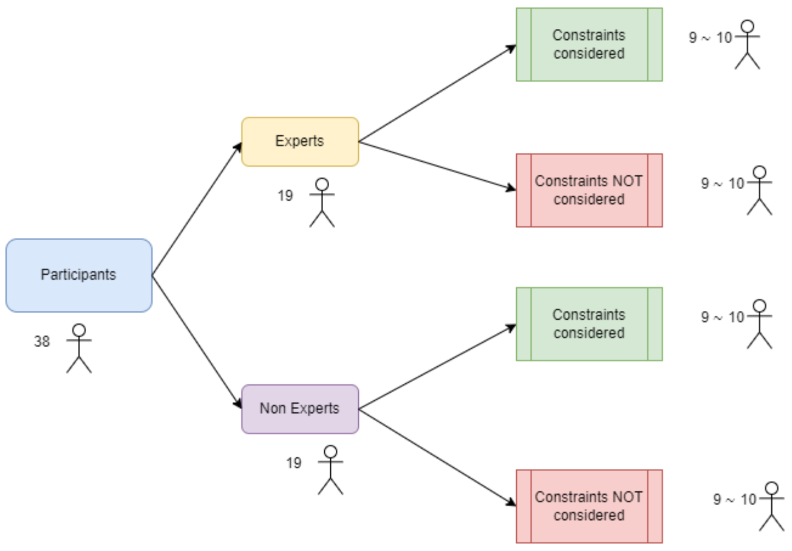


Figure 4.3: Number and organization of participants

Recruitment The participants were recruited by posts on the website of the university or in the Laboratory office.

4.2 Experiment description

4.2.1 Participants tasks

The participants will all carry out a simple assembly task with the cobot as training. The task will be presented to the participants as "chemistry with Duplo blocks".

All participants will wear surgical gloves to move the blocks, with the exception of the red blocks that are to be moved with a different special glove.

The participants will either be Experts or Non experts. The first group, the Experts, will work on the complex assembly task three times alone, without the assistance of the robot. The second group, the Non Experts, will not train on the complex task alone, but will directly work in collaboration with the cobot so that they will see the task for the first time in this occasion.

At first, all participants will complete a small and simple "training" task shown in in figure 4.6, together with the cobot, to become acquainted with the movements and with the system and not be frightened. Afterwards, all of the participants will work on the complex task with the cobot. The participant must reproduce the complex model in figure 4.7 with the cobot without any time constraint known to them, while answering simple math additions to increase the cognitive load.

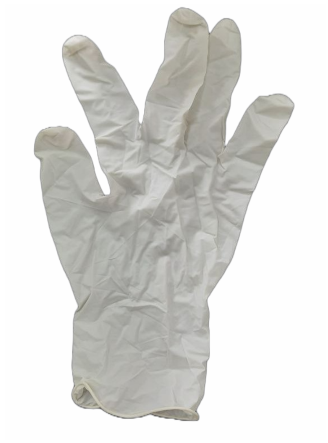


Figure 4.4: Latex glove

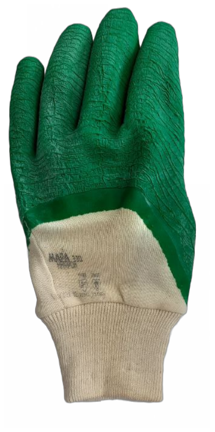


Figure 4.5: Special glove

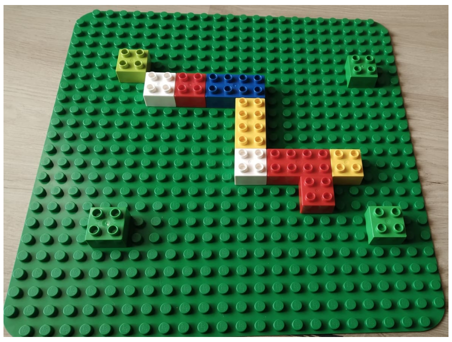


Figure 4.6: Simple task

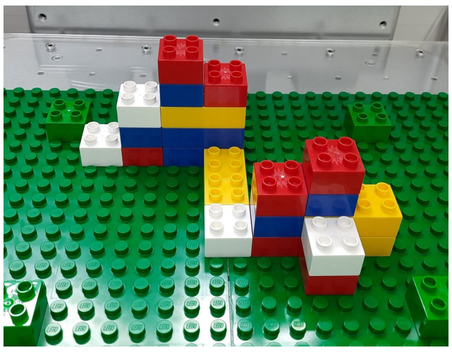


Figure 4.7: Complex task

In both conditions, the participant will have access to more than half of the Duplo's stock to complete the task. The cobot will have access to the remaining of the Duplo's stock and will have access to the model to complete.

The blocks in the lower part in figure 4.8 are reserved for the human while the blocks in the upper most part are respectively for the left and right arm of the cobot.

Presentation of the experience

« The experiment will last about 1 hour. You will perform an assembly task in collaboration with a cobot, this is not evaluative, do your best without pressure. Before this task, we will make an easy model to train you. Then, the goal for you is to complete the model I will show you in a few moments with a stock Duplo. Our

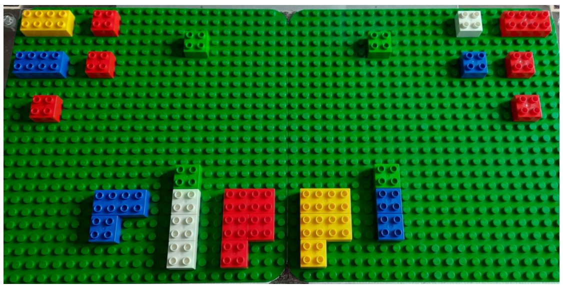


Figure 4.8: Complex distribution stock

goal is to better understand the effect of collaboration on performance and gestures. So, we're going to film your upper body movements. These videos will allow us to count the number of gestures you make, the number of errors, etc. Once we have collected this information, we will delete these videos. Do you agree with this? After the Duplo assembly task, you will fill out a quick 10-minute questionnaire on a computer provided for this purpose. »

Presentation for the experts

*« You will have a supply of Duplo parts (*show participant's supply*) and the model to make it. The cobot will have a supply of Duplo parts (*show cobot's supply*). You are not allowed to take the cobot's stock.*
The model must be made inside the green square represented by the 4 green pieces. Every 15 seconds, I will ask you for the solution of a simple addition, you must give me the right answer otherwise I will continue to ask for the answer with insistence. Do it as fast as you can and avoid making mistakes. You will start as soon as you are ready and you will tell me "I'm done" when you are finished so that I can stop the recording. »

Chapter 5

Experimental Results

The results of the experiment conducted will be reported in this chapter. The analysis of the results was made in collaboration with a work psychology and ergonomy expert, and the results presented in this thesis are preliminary results that will be presented in more formal contexts in the future.

5.0.1 Description of the conditions

Two different conditions are studied: adaptability degree of the cobot and Expertise level. Each one of these conditions have 2 degrees to it.

Adaptation to constraints of the cobotic system

The condition "Adapted" and "Not Adapted" that we will explore in this chapter refer to the degree of adaptation to constraints applied to the cobotic system in the experiment.
Not Adapted: The cobotic system is unaware of the human constraint of having to wear a special glove to handle red blocks. The cobot plans and performs its actions without giving more importance to any block specifically.
Adapted: The cobotic system is aware of the human constraint of having to wear a special glove to handle red blocks. The cobot plans and performs its actions assigning priority and more weight to the manipulation of red blocks, so that the human is freed of the burden of managing more red blocks.

Half of the participants (19 participants) completes the task with the degree of adaptation "Adapted" and the other half will work in the "Not Adapted" alternative.

Expertise level of the human operator

The condition "Expert" and "Not Expert" that we will explore in this chapter refer to the level of espertize of the human regarding the completion of the task by themselves.
Expert: A participant who trained on the complex assembly task three times alone, without the assistance of the robot, to learn the task pattern and the use of the glove.
Not Expert:A participant who works on the complex assembly task for the first time with the help of the robot. They have no previous knowledge of the task pattern, only a picture is showed while the completion of the task is happening.

Half of the participants (19 participants) completes the task being "expert" and the other half will be "not expert".

5.0.2 Description of the criteria of evaluation

The criteria considered in the evaluation are:

Performances:

- Number of gestures : number of limb movements above the workstation
- Number of errors: cube misplacement, rule breaks, misuse of the gloves
- Completion time

Risk taking or delaying:

- Number of red blocks handled
- Level of first usage of the special glove

Questionnaire assessed parameters:

- Acceptability: usability perceived, coherence perceived, pleasure perceived, usefulness perceived, social influence, trust
- Workload

5.1 Expected results

Hypothesis 1: The adaptive condition of the cobotic system will have an effect on the participant's performance (completion time, errors, gestures), on the risks taken (number of dangerous blocks handled) and on the delay of usage of the special glove.

This means that we would expect the group of participants in the "Adapted" condition,in comparison with the "Not adapted" group to:

- Report less gestures
- Be faster, have a lower completion time
- Make less errors
- Take less security risks managing less red blocks
- Delay the use of the security glove to the last levels of the task

Hypothesis 2: The adaptive cobotic system will result in reduced perceived workload and perceived acceptability(usability perceived, coherence perceived, pleasure perceived, usefulness perceived, social influence and trust), for the human participants.

We would expect the "Adapted" group to perceive the robot as more useful and acceptable, and to report a lower workload score in their questionnaire.

Hypothesis 3: The operator's expertise (expert vs non-expert) will have no effect on the acceptability (usability perceived, coherence perceived, pleasure perceived, usefulness perceived, social influence and trust). The operator's expertise will have an effect on the workload, on the performance (number of errors, time and number of gestures), on the risk taken and on the security constraints of the cobotic system.
We would aim for the "Expert" group to perceive the system in a similar way as the "Non- expert" group because that would mean that the system is equally acceptable in both conditions.
Since the system used by both groups is the same and there are no changes in the cobotic system between one group and the other, the ideal scenario would be one where the perception of the system from the participants is not influenced by the level of expertise.

We would also expect the operator's expertise to lower the number of errors, time and number of gestures, and in general experts should have better performances than non experts.

5.2 Results for the adaptation to constraints of the cobotic system

Objective performance

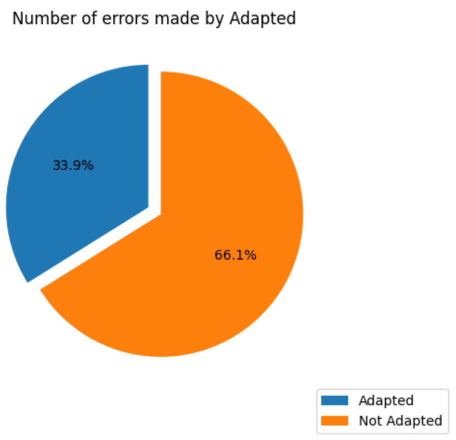


Figure 5.1: Analysis of number of errors by adapted condition

Adaptation	# errors
Yes	21
No	41
Total Errors	62

Table 5.1: Number of errors - Adaptation condition

The adaptation of the cobotic system to human constraints has an effect on the number of errors that the participant makes.
The majority of errors happened in the "Not adapted" condition. The group that was assisted more by the robot made only 1/3 of the total errors, while the group that was not assisted made 2/3 of the total errors.
As it is shown, **the adaptation of the cobot helps the participant commit less errors**.

Adaptation	Avg Time (seconds)
Yes	319.578947
No	346.736842

Table 5.2: Average time in seconds - Adaptation condition

Visually, as we see in table 5.2, the average time is slightly better in the "Adapted" case but its significance is insufficient to draw any meaningful conclusions. This is confirmed by the T-test: T=-0,972550239, p=0,169. Since p>0.05 the T-test is non-significant.
Time completion is not affected by the adaptation of the cobot to constraints

As we can see from table 5.3, the average number of gestures does not vary particularly depending on the "Adaptation" condition. This is confirmed by the T-test: T=-0,6787, p=0,250. Since p>0.05 the T-test is nonsignificant.
Number of gestures is not affected by the adaptation of the cobot to constraints

Adaptation	Avg # Gestures
Yes	28.21
No	30.00

Table 5.3: Average number of gestures - Adaptation condition

Risk taking

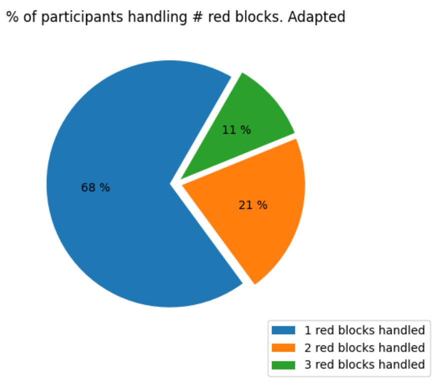


Figure 5.2: Analysis of number of red blocks handled by adapted condition

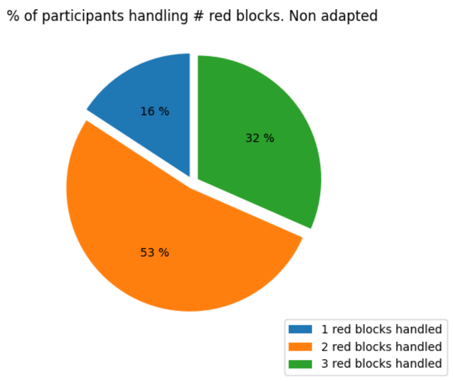


Figure 5.3: Analysis of number of red blocks handled by non adapted condition

The adaptation of the cobotic system to human constraints has a strong effect on the number of risks that the participant takes, which for the chosen constraints it is equal to the number of toxic red blocks that the human choose to manage. The participants were obliged to manage at least one red block and at worse 3 blocks.
As it is shown in the graphs above, in figure 5.2 , in the "Adapted" condition most of the participants (70%) only choose to handle 1 red block, which is the minimum required. When the participants notice that the robot helps them manage the red blocks, they did not take more risks than it was necessary. 21% of this category

managed 2 blocks, one more than it was necessary, and only 11% managed all 3 dangerous blocks.
In the "Not adapted" condition in figure 5.3, instead, we notice that only a very small percentage of participants (16%) took only the necessary risk of handling 1 red block, while the remaining 84% took more risks than it was necessary, managing the toxic blocks either 2 times (52% of the participants of the "not adapted" condition) or 3 times in the 32% of cases.

It is clear that **the participants that experienced the "Not adapted" condition felt the need to take more risks while most of the participants of the "Adapted" condition choose to only take the risks that were strictly necessary.**

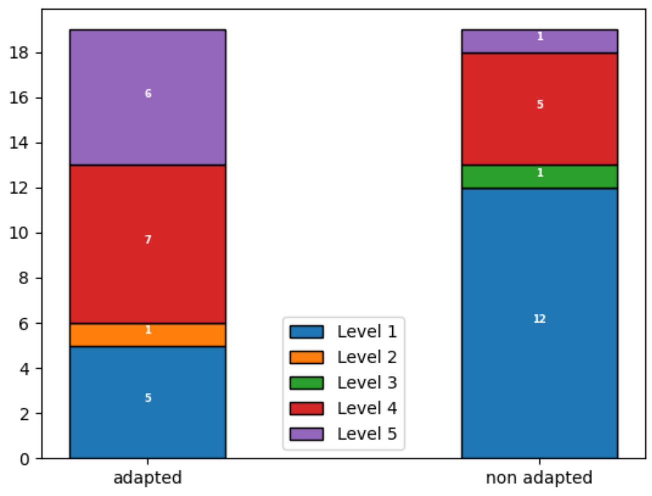


Figure 5.4: Level of first use of the special glove: number of participants

The adaptation of the cobotic system to human constraints has a strong effect on the delay of the use of the special glove and managing of the first toxic block, that corresponds to the delay of risk taking to when it feels necessary to the participant. In the "Adapted" condition, as we can see from figure 5.4 most of the participants use the special glove for the first time during the last 2 levels (level 4 and 5), actively delaying the risk taking when they notice that the robot has (or will have) no red blocks left to help.

Only 5 participants used the special glove from the first level. In the "Not adapted" condition, the majority of participants (12 out of 19 participants) felt the necessity to use the special glove from the first level, starting to take the risks since

the beginning of the experiment. In particular, only 1 participant delayed the risk until the last level compared to the 6 participants in the "adapted" case.
This proves that **the "Adapted" condition effectively delays the use of the special glove, which means that it delays taking the risk to when it is strictly necessary.**

Self assessed perception of the system from questionnaires

The table below presents for each evaluated criteria recorded through questionnaire the average scores and the standard deviation. As we can see, we notice no relevant differences in the results of the "Adapted" class and the "Not adapted" class. Also the workload perceived is similar in both classes.

Average scores of self assessed parameters (from questionnaires)

Condition	Usability	Coherence	Pleasure	Usefulness	Influence	Trust	Workload
Adapted	M=21,21 (SD=4,48)	M=16,63 (SD=4,27)	M=26,37 (SD=8,1)	M=17,74 (SD=5,14)	M=13,05 (SD=4,7)	M=21,26 (SD=5,37)	120,66 (SD=62,1)
Non-adapted	M=21,95 (SD=3,96)	M=16,47 (SD=3,5)	M=28,05 (SD=6,1)	M=17,21 (SD=7,11)	M=13,89 (SD=4)	M=20,58 (SD=4,69)	M=129,95 (SD=64,2)

Table 5.4: Self assessed parameters - Adaptation condition

These results in table 5.4 are interesting because it shows that the cobot was not perceived differently by the participants of the two classes. This means that in both cases the cobot was able to provide a smooth collaboration. The adaptation to the constraints was executed in a way that did not interfere with the human perception of the cobotic system, but providing the expected practical benefits such as limiting errors and risks and delaying the risks that the human takes.

5.3 Results for the operator's expertise level

Objective performance

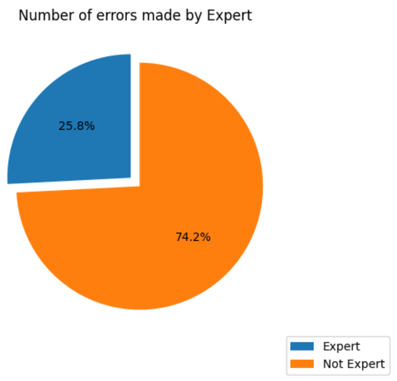


Figure 5.5: Analysis of number of errors by expert condition

Expertise	# errors
Yes	16
No	46
Total Errors	62

Table 5.5: Number of errors - Expertise condition

The Expertise level of the participant on the task has an effect on the number of errors that the participant makes.
The majority of errors happened in the "Not expert" condition. The Expert group made only 1/4 of the total errors, while the not expert group made 3/4 of the total errors.
For this, it was proven that **the Expert participants commit less errors**.

Expertise	Average Time (seconds)
Yes	316.157895
No	350.157895

Table 5.6: Average time in seconds - Expertise condition

Similarly as in the Adapted conditions, also for the Expert condition the average time is slightly better in the "Expert" case but its significance is not sufficient to draw any meaningful conclusions. **Time completion is not particularly affected by the expertise level of the participant**

As we can see from table 5.7, the average number of gestures only slightly varies depending on the "Expert" condition. It is not enough to draw any meaningful conclusions.
Number of gestures is not particularly affected by the expertise level of the participant

Expertise	Avg # Gestures
Yes	27.05
No	31.15

Table 5.7: Number of gestures - Expertise condition

Self assessed perception of the system from questionnaires

Table 5.8 presents for each evaluated criteria recorded through questionnaire the average scores and the standard deviation. We notice no relevant differences in the results of the "Expert" class and the "Not Expert" class. Also the workload perceived is similar in both classes.

Average scores of self assessed parameters (from questionnaires)

Condition	Usability	Coherence	Pleasure	Usefulness	Influence	Trust	Workload
Expert	M=21.95 (SD=3.82)	M=16,84 (SD=3,69)	M=27,42 (SD=6,27)	M=17 (SD=5,99)	M=13,74 (SD=4,16)	M=20,16 (SD=4,6)	M=20,16 (SD=4,6)
Non-expert	M=21.21 (SD=4.6)	M=16,26 (SD=4,09)	M=27,8 (SD=8,08)	M=17,95 (SD=6,39)	M=13,21 (SD=4,55)	M=21,68 (SD=5,37)	M=21,68 (SD=5,37)

Table 5.8: Self assessed parameters - Expertise condition

Even if from the analysis of the objective performance we can see that the condition expert/non-expert worked (less errors, less risks taken) it did not impact the evaluation of the cobot made by the participants in the quesionnaires.
In the literature, such as in [31], expert participants usually perceived the cobot's help, its usefulness, trust, and all the other parameters evaluated as less valuable than their non-expert counterpart. In this analyzed case, instead, the perception of the expert- not expert did not change, but their performance did.
This means that **the framework was able to fill the gap between the Expert group and the Non expert group regarding the perception of the cobot and both groups perceived it in the same way.**
The cobot has successfully adapted to the"expertise" human variability .

5.4 Participant's answers to the acceptability of the cobot- questionnaire

As we have seen in the previous chapters, the constraints and the human variabilities have no visible effect in the answers that the participants gave to the questionnaires. In the following table we report the average answers of the participants for the "Acceptability of the cobot" questionnaire. The participants answered 23 questions that were made to evaluate the Perceived Usability, Coherence, Pleasure, Utility, Social influence and Trust in the cobotic system that they had juts collaborated with. The participants were asked to select the grade of agreement with some written affirmations. They were able to choose a score from 1 to 7 where: 1 - Strongly disagree; 2 - Disagree; 3 - Somewhat disagree; 4 - Neither agree nor disagree; 5 - Somewhat agree; 6 - Agree; 7 - Completely agree

Average answers for Acceptability questionnaire
Scores from 1- (Completely disagree) to 7- (Completely agreee)

Usability :	
Understanding how to collaborate with a cobot was clear to me.	6,10
Working with a cobot did not require much mental effort	4,71
I found working with a cobot easy.	5,65
I was able to easily complete the task in my own way thanks to the cobot.	5,10
Coherence :	
The collaboration with a cobot was relevant to the content of the tasks that I carried out.	5,39
For the tasks to be performed, collaboration with a cobot was appropriate.	5,47
The collaboration with a cobot was in line with the tasks that I carried out.	5,68
Pleasure :	
Because using a cobot was very fun.	5,5
Because using a cobot was a lot of fun for me.	5,47
Because thanks to the cobot I was able to satisfy my curiosity to discover new things.	5,95
Because I had a lot of fun interacting with a cobot.	5,36
Because the cobot allowed me to learn/do things that I find very interesting.	4,92
Utility :	
Collaborating with a cobot improved my performance in the task.	3,87
Collaborating with a cobot during this task increased my performance without making me more tired	4,39
Collaborating with a cobot has improved my efficiency in the task.	4,34
I find the collaboration with a cobot helpful in the task.	4,87
Social influence :	
People whose opinion I value might encourage me to reuse the cobot to complete the task.	4,63
The people around me who I am used to listening to could advise me to reuse the cobot to carry out the task.	4,34
The people around me who are important to me could encourage me to reuse the cobot.	4,5
Trust :	
A cobot seems to provide reliable information.	5,52
I have confidence in the information that a cobot could provide.	5,23
The cobot seems reliable.	5,29
I think you can trust a cobot.	4,87

Table 5.10: Acceptability questionnaire, average answers

The participants, on average, they agreed the robot was easy to work with and to understand how to use it. They also agree on the coherence of the cobot, finding it appropriate for the realized task. The participants enjoyed working with the robot and found it interesting.
Some lower scores were registered for the utility and the social influence.
The average score tells us that the participants, on average, neither agree nor disagree with the affirmations about social influence and they do not believe the robot was particularly useful for them to perform the task.
In other words, they do not particularly believe that the robot gave them essential and valuable help, which for the task we selected it is reasonable, since the participants could have easily managed all the blocks themselves.
To conclude, most of the participants found the cobot somewhat trustworthy.

Chapter 6

Conclusion

In this study we focused on the problem of automatic task planning for adapting a cobotic system to specific human factors and constraints in the context of human-robot collaboration (HRC). The increasing presence of robots in industrial settings has sparked interest in HRC, and this thesis aimed to research on the need for human freedom and adaptability while considering human constraints and promoting ergonomic choices.

A framework was proposed to tackle these problems, incorporating a dynamic adaptive planner based on PDDL, a perception system to detect human actions, and a graphic user interface for communication.
An experiment was conducted to validate the proposed implementations and assess their usefulness. The experiment involved a collaborative task between humans and robots, evaluating objective performance, risk taking, workload, and acceptability.

The results of the experiment demonstrated the effectiveness of the adaptive cobotic system in addressing human constraints and improving the overall human experience. In terms of performance, the adaptation of the cobot significantly reduced the number of errors made by participants, showcasing its ability to handle unexpected human behaviors and constraints. While there were no significant differences in completion time and the number of gestures between the adapted and non-adapted conditions, the adaptive system showed a clear impact on reducing the risks taken by participants.

The experiment also considered human variability, including cognitive variability and expertise level. It was observed that expertise played a role in performance, with expert participants committing fewer errors compared to non-experts. However, the adaptability of the cobotic system effectively bridged the gap between experts and non-experts in terms of their perception of the cobot's assistance.

The participant's assessment of the cobot's acceptability through questionnaires revealed positive feedback in terms of usability, coherence, pleasure, and interest. While the participants did not strongly agree on the robot's utility and social influence, it can be attributed to the task's nature, where participants could have managed the blocks themselves without significant difficulty.

Overall, the results support the effectiveness of the proposed framework for automatic task planning in HRC. The adaptability of the cobotic system positively impacted performance, risk management and user experience. This research contributes to the field of HRC by addressing the challenges of human freedom, adaptability, and ergonomics, ultimately paving the way for safer and more efficient collaboration between humans and robots in industrial settings.

Future work:

While this thesis has focused on the problem of automatic task planning for adapting a cobotic system to specific human factors, there are several avenues for future research and development in this area. The proposed framework and experiment have provided valuable insights and demonstrated the adaptability and effectiveness of the cobotic system. Building upon these findings, the following areas offer opportunities for further exploration and improvement:

Dynamic Task Adaptation: Currently, the task planning in the cobotic system is based on a predefined goal configuration of Lego Duplo blocks. In future work, the task can be made more dynamic, allowing it to change every time. This could involve introducing variability in the initial stock of blocks that the robot and human have to manipulate, and on the goal state of the task. By dynamically adapting the task requirements, the cobotic system can provide more flexibility and responsiveness to evolving work scenarios.

Dynamic PDDL domain for a more responsive Planner: The proposed framework utilizes a dynamic adaptive planner based on PDDL (Planning Domain Definition Language) where the Problem is dynamic, to handle unexpected human behaviors and constraints. To enhance the system's adaptability further, future work can focus on developing a dynamic planner that takes into account initial user inputs to generate the PDDL domain. By incorporating an initial series of questions in the graphic user interface (GUI) about the task and human constraints, the planner can dynamically adjust its planning strategies from the outset, leading to more efficient and personalized task allocation.

User-Centric Interface: The graphic user interface (GUI) plays a crucial role in facilitating effective human-robot collaboration. In future work, improvements can be made to the interface design to enhance user experience and promote seamless communication between the human and robot. The interface can provide real-time feedback on the robot's perception of the environment and the planned actions, enabling the human to make informed decisions about when to activate the robot for assistance. Moreover, incorporating user preferences and customization options in the interface can further enhance the usability and acceptability of the cobotic system.

Integration of Machine Learning: The inclusion of machine learning techniques can greatly enhance the adaptability and intelligence of the cobotic system. By leveraging machine learning algorithms, the system can learn from user interactions and continuously improve its performance over time. For example, the system can learn to predict user preferences and adapt its planning strategies accordingly. Additionally, machine learning can aid in recognizing and adapting to new human behaviors and constraints, making the cobotic system even more adaptable and efficient.

Pose Detection for Ergonomics: To enhance the ergonomic aspects of human-robot collaboration, future work can focus on integrating human limb pose detection techniques into the cobotic system. By accurately tracking and analyzing the position and orientation of the human limbs, the system can adapt its movements and assistive actions to ensure optimal ergonomics for the human operator. This can help prevent strain, fatigue, and potential injuries caused by repetitive or awkward movements. By combining limb pose detection with the dynamic planner mentioned earlier, the cobotic system can proactively adjust its assistance strategies based on the detected limb poses, providing personalized and ergonomic support to the human operator. This integration of ergonomics and adaptive movement planning will contribute to the overall well-being and safety of the human operator, fostering a more harmonious and efficient human-robot collaboration environment.

Real-World Industrial Implementations: While the experiment conducted in this thesis provides valuable insights, future work should focus on validating the proposed framework in real-world industrial settings. By deploying the cobotic system in actual manufacturing or assembly environments, it will be possible to assess its performance, adaptability, and usability in practical scenarios. This can involve collaborating with industry partners and conducting extensive field trials to evaluate the system's effectiveness and identify areas for further refinement.

In conclusion, the proposed framework and experimental results have demonstrated the adaptability of the cobotic system, improved human experience, and effective planning. Future work in dynamic task adaptation, and real-world industrial implementations will contribute to further advancements in this field, ultimately paving the way for more efficient and productive human-robot collaboration in industrial settings.

Bibliography

[1] Francesco Chiacchio, Georgios Petropoulos, and David Pichler. «The impact of industrial robots on EU employment and wages: A local labour market approach. Bruegel Working Paper Issue 02/18 April 2018». In: 2018 (cit. on p. 1).

[2] Asmita Bisen and Himanshu Payal. «Collaborative robots for industrial tasks: A review». In: *Materials Today: Proceedings* 52 (Oct. 2021). DOI: `10.1016/j.matpr.2021.09.263` (cit. on p. 1).

[3] Diego Rodríguez-Guerra, Gorka Sorrosal, Cabanes I., and Carlos Calleja. «Human-Robot Interaction Review: Challenges and Solutions for Modern Industrial Environments». In: *IEEE Access* PP (July 2021), pp. 1–1. DOI: `10.1109/ACCESS.2021.3099287` (cit. on p. 1).

[4] Sehoon Kim. «Working With Robots: Human Resource Development Considerations in Human-Robot Interaction». In: *Human Resource Development Review* 21.1 (2022), pp. 48–74. DOI: `10.1177/15344843211068810`. eprint: `https://doi.org/10.1177/15344843211068810`. URL: `https://doi.org/10.1177/15344843211068810` (cit. on pp. 1–4).

[5] J. Edward, Witaya Wannasuphoprasit, and Michael Peshkin. «Cobots: Robots For Collaboration With Human Operators». In: (Mar. 1999) (cit. on p. 2).

[6] Alessio Baratta, Antonio Cimino, Maria Grazia Gnoni, and Francesco Longo. «Human Robot Collaboration in Industry 4.0: a literature review». In: *Procedia Computer Science* 217 (2023). 4th International Conference on Industry 4.0 and Smart Manufacturing, pp. 1887–1895. ISSN: 1877-0509. DOI: `https://doi.org/10.1016/j.procs.2022.12.389`. URL: `https://www.sciencedirect.com/science/article/pii/S1877050922024747` (cit. on p. 3).

[7] Sofya Langman, Nicole Capicotto, Yaser Maddahi, and Kourosh Zareinia. «Roboethics principles and policies in Europe and North America». In: *SN Applied Sciences* 3.12 (2021), p. 857. ISSN: 2523-3971. DOI: `10.1007/s42452-021-04853-5`. URL: `https://doi.org/10.1007/s42452-021-04853-5` (cit. on p. 3).

[8] Margaret Boden et al. «Principles of robotics: regulating robots in the real world». In: *Connection Science* 29 (Apr. 2017), pp. 124–129. DOI: `10.1080/09540091.2016.1271400` (cit. on p. 3).

[9] Mohammed Khesbak. «Enabling Collision Avoidance in Human Robot Collaborative Industrial Environment: Prototype of Safe UWB-based Hybrid Positioning System Use case: Collaborative assembly of power transformers». PhD thesis. Aug. 2020 (cit. on p. 3).

[10] Sami Haddadin. *Towards Safe Robots: Approaching Asimov's 1st Law.* Springer Publishing Company, Incorporated, 2013. ISBN: 3642403077 (cit. on p. 3).

[11] Janet I. Minshew. «32 - Ergonomics». In: *Theory and Practice of Histological Techniques (Sixth Edition).* Ed. by John D. Bancroft and Marilyn Gamble. Sixth Edition. Edinburgh: Churchill Livingstone, 2008, pp. 661–671. ISBN: 978-0-443-10279-0. DOI: `https://doi.org/10.1016/B978-0-443-10279-0.50039-7`. URL: `https://www.sciencedirect.com/science/article/pii/B9780443102790500397` (cit. on pp. 3, 5).

[12] Richard E. Fikes and Nils J. Nilsson. «Strips: A new approach to the application of theorem proving to problem solving». In: *Artificial Intelligence* 2.3 (1971), pp. 189–208. ISSN: 0004-3702. DOI: `https://doi.org/10.1016/0004-3702(71)90010-5`. URL: `https://www.sciencedirect.com/science/article/pii/0004370271900105` (cit. on pp. 5, 13).

[13] Malik Ghallab et al. «PDDL - The Planning Domain Definition Language». In: (Aug. 1998) (cit. on pp. 5, 8, 13).

[14] Nancy N. Byl, Mary F. Barbe, Carolyn Byl Dolan, and Grant Glass. «Chapter 27 - Repetitive Stress Pathology: Soft Tissue*The authors, editors, and publisher wish to acknowledge Ann Barr for her contributions on this topic in the previous edition.» In: *Pathology and Intervention in Musculoskeletal Rehabilitation (Second Edition).* Ed. by David J. Magee, James E. Zachazewski, William S. Quillen, and Robert C. Manske. Second Edition. W.B. Saunders, 2016, pp. 938–1004. ISBN: 978-0-323-31072-7. DOI: `https://doi.org/10.1016/B978-0-323-31072-7.00027-0`. URL: `https://www.sciencedirect.com/science/article/pii/B9780323310727000270` (cit. on p. 5).

[15] Marta Lorenzini, Marta Lagomarsino, Luca Fortini, Soheil Gholami, and Arash Ajoudani. «Ergonomic human-robot collaboration in industry: A review». In: *Frontiers in Robotics and AI* 9 (2023). ISSN: 2296-9144. DOI: `10.3389/frobt.2022.813907`. URL: `https://www.frontiersin.org/articles/10.3389/frobt.2022.813907` (cit. on p. 5).

[16] Wansoo Kim et al. «Adaptable Workstations for Human-Robot Collaboration: A Reconfigurable Framework for Improving Worker Ergonomics and Productivity». In: *IEEE Robotics & Automation Magazine* 26.3 (2019), pp. 14–26. DOI: 10.1109/MRA.2018.2890460 (cit. on p. 5).

[17] Michael Gelfond and Vladimir Lifschitz. «Action Languages». In: *Electron. Trans. Artif. Intell.* 2 (1998), pp. 193–210 (cit. on p. 13).

[18] Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl. «GOLOG: A logic programming language for dynamic domains». In: *The Journal of Logic Programming* 31.1 (1997). Reasoning about Action and Change, pp. 59–83. ISSN: 0743-1066. DOI: https://doi.org/10.1016/S0743-1066(96)00121-5. URL: https://www.sciencedirect.com/science/article/pii/S0743106696001215 (cit. on p. 13).

[19] Ken Currie and Austin Tate. «O-Plan: The open planning architecture». In: *Artificial Intelligence* 52.1 (1991), pp. 49–86. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(91)90024-E. URL: https://www.sciencedirect.com/science/article/pii/000437029190024E (cit. on p. 13).

[20] Sergio Jiménez, Tomas De la Rosa, Susana Fernandez, Fernando Fernández, and Daniel Borrajo. «A review of machine learning for automated planning». In: *The Knowledge Engineering Review* 27 (Dec. 2012). DOI: 10.1017/S026988891200001X (cit. on p. 13).

[21] Einollah Pira. «Using deep learning techniques for solving AI planning problems specified through graph transformations». In: *Soft Computing* 26.22 (2022), pp. 12217–12234. ISSN: 1433-7479. DOI: 10.1007/s00500-022-07044-5. URL: https://doi.org/10.1007/s00500-022-07044-5 (cit. on p. 13).

[22] Ilche Georgievski and Marco Aiello. «HTN planning: Overview, comparison, and beyond». In: *Artificial Intelligence* 222 (2015), pp. 124–156. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2015.02.002. URL: https://www.sciencedirect.com/science/article/pii/S0004370215000247 (cit. on p. 14).

[23] Kai Li, Quan Liu, Wenjun Xu, Jiayi Liu, Zude Zhou, and Hao Feng. «Sequence Planning Considering Human Fatigue for Human-Robot Collaboration in Disassembly». In: *Procedia CIRP* 83 (2019). 11th CIRP Conference on Industrial Product-Service Systems, pp. 95–104. ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2019.04.127. URL: https://www.sciencedirect.com/science/article/pii/S2212827119307413 (cit. on p. 15).

[24] Shunsuke Mochizuki, Yosuke Kawasaki, and Masaki Takahashi. «ProTAMP: Probabilistic Task and Motion Planning Considering Human Action for Harmonious Collaboration». In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 2159–2166. DOI: 10.1109/IROS47612.2022.9982074 (cit. on p. 15).

[25] Lars Johannsmeier and Sami Haddadin. «A Hierarchical Human-Robot Interaction-Planning Framework for Task Allocation in Collaborative Industrial Assembly Processes». In: *IEEE Robotics and Automation Letters* 2.1 (2017), pp. 41–48. DOI: 10.1109/LRA.2016.2535907 (cit. on p. 16).

[26] Alessandro Roncone, Olivier Mangin, and Brian Scassellati. «Transparent role assignment and task allocation in human robot collaboration». In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1014–1021. DOI: 10.1109/ICRA.2017.7989122 (cit. on p. 16).

[27] Hossein Karami, Kourosh Darvish, and Fulvio Mastrogiovanni. «A Task Allocation Approach for Human-Robot Collaboration in Product Defects Inspection Scenarios». In: *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2020, pp. 1127–1134. DOI: 10.1109/RO-MAN47096.2020.9223455 (cit. on p. 16).

[28] Giulia Bruno and Dario Antonelli. «Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells». In: *The International Journal of Advanced Manufacturing Technology* 98 (Oct. 2018). DOI: 10.1007/s00170-018-2400-4 (cit. on p. 17).

[29] Ya Liu, Zhigang Jiang, and Chao Ke. «A Task Allocation Method in Human-Robot Collaboration (HRC) for the Disassembly of Automotive Traction Batteries». In: *2022 International Conference on Cyber-Physical Social Intelligence (ICCSI)*. 2022, pp. 332–337. DOI: 10.1109/ICCSI55536.2022.9970619 (cit. on p. 17).

[30] M. Helmert. «The Fast Downward Planning System». In: *Journal of Artificial Intelligence Research* 26 (July 2006), pp. 191–246. DOI: 10.1613/jair.1705. URL: https://doi.org/10.1613/jair.1705 (cit. on p. 27).

[31] E. Cippelletti. «Aide a la conception, test de l'usage et de l'acceptation d'un logiciel de maintenance». Doctoral dissertation. Universite Grenoble Alpes (ComUE), 2017 (cit. on p. 51).