FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Human-Machine Interaction design in Cyber-Physical Production Systems

Henrique Castello Branco Lopes Cardoso

U. PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado em Engenharia de Software

Supervisor: Gil Gonçalves

Co-Supervisor: João Reis

July 22, 2019

Human-Machine Interaction design in Cyber-Physical Production Systems

Henrique Castello Branco Lopes Cardoso

Mestrado em Engenharia de Software

July 22, 2019

Abstract

Cognitive overload affects the way we think, react and reduces our reasoning capacities. As the day goes by we feel more exhausted and it reflects in our performance along the day. The capability to manage our own cognitive load is fairly important for our well being and with the technology available today we are able to monetize our fatigue and stress levels. With this technology, we are able to control our workflow and learn to manage our tiredness. With many studies done on this area, there were discovered and developed some techniques that help to reduce the cognitive load by addressing the way we live and the way we interact with technology without the need to equip any sensor tracker.

Every job in the world demands some sort of mental or physical effort in order to be able to do it correctly, yet, some of them are more demanding due to the type of activities they require. Production and assembly jobs are usually the ones that require a higher level of focus from the employees, as besides the activity itself being almost the same throughout the whole working shift, which in most cases is longer that the average found on other jobs, the margin for error is minimal as it could cause to serious impact on the company's goals.

As factories thrive to produce better and more precise products, the cognitive load increases and the factory workers performance start to affect the results. One of the multiple aspects, detected by the PSA Group, that is not as evolved as it could be and causes an unnecessary overload on the operator, is related with the way the information is disposed. Nowadays, information regarding the tasks and assets needed to assemble a vehicle, is transmitted using a sheet of paper that covers all of the assembly stations, regardless of who reads it. This method treats every operator equally, not taking into consideration that no two people are alike.

As a way to approach this problem, this dissertation presents a possible solution that aims to help the assembly line workers by reducing their cognitive load when looking for the tasks they need to execute for each product and consequently help the business. The proposed solution focuses on a support interface designed to draw the worker's attention to the most important information.

With this dissertation, and within the proposed testing scenarios, we were able to conclude that an improved and adaptive interface can positively impact the performance on people, and therefore, impact the operator's performance when assembling products. Still there is a lot from where to improve this approach which can result in a even larger discrepancy.

Keywords: Support Interface, Cognitive Load, Expertise level, CPS

Resumo

A sobrecarga cognitiva afeta a maneira como pensamos, reagimos e reduz a nossa capacidade de raciocínio. Ao longo do dia, é crescente o cansaço e isso reflete-se no desempenho de atividades. Conseguir gerir essa carga cognitiva é importante para o bem-estar físico e psicológico. Com a tecnologia atualmente disponível, é possível monitorizar os níveis de stress e fadiga. Com esta tecnologia, somos capazes de controlar o fluxo de trabalho e aprender a gerir o cansaço. Com os vários estudos feitos nesta área, foram descobertas e desenvolvidas várias técnicas que ajudam a reduzir a carga cognitiva condicionando a maneira como vivemos e interagimos com a tecnologia, sem que seja necessário a utilização de nenhum sensor de monitorização.

Independentemente do trabalho, todos exigem algum tipo de esforço mental ou físico para que seja realizado corretamente, contudo note-se que alguns trabalhos são mais exigentes do que outros, devido ao tipo de atividades implícitas. Trabalhos relacionados com a produção e montagem, são, normalmente os que exigem um nível de concentração mais alto, porque, apesar da sua atividade ser muito repetitiva, os turnos são de elevada duração e a margem de erro é baixa, podendo ter um grande impacto no sucesso da empresa.

Como as fábricas ambicionam produzir mais e melhores produtos, a carga cognitiva dos trabalhadores aumenta e o seu desempenho começa a afetar os resultados. Um dos muitos aspetos detetados pelo Grupo PSA, que não está muito desenvolvido, e provoca uma desnecessária sobrecarga no operador, está relacionado com a maneira como a informação é apresentada. Hoje em dia, a informação relacionada com as tarefas e ativos necessários para montar um veículo, é transmitida através uma folha de papel que percorre todas as estações de montagem, independentemente do operador que está a trabalhar no momento. Este método trata todos os agentes a mesma maneira, não tendo em consideração a especificidade de cada um.

Evidenciado o problema, esta dissertação apresenta uma possível solução que procura ajudar um trabalhador da linha de montagem. Pretende-se reduzir o seu esforço cognitivo na leitura das tarefas, nomeadamente acerca do que precisa de completar para cada produto e consequentemente melhorar os resultados finais. A solução proposta foca-se numa interface de suporte desenhada para dirigir a atenção dos trabalhadores para a informação mais relevante.

Com esta dissertação, e com os testes executados, fomos capazes de concluir que uma interface adaptativa pode ter um impacto positivo no desempenho das pessoas e assim, melhorar o desempenho dos trabalhadores aquando da montagem do produto. Ainda assim, existe muita coisa a ser melhorada nesta abordagem, o que se pode repercutir em resultados ainda melhores.

Keywords: Interface de Suporte, Carga cognitiva, Nível de experiência, CPS

"The people who are crazy enough to think they can change the world,
are the ones who do."

- Steve Jobs

Contents

List of Figures

List of Tables

Abbreviations

AUI	Adaptive User Interface
CL	Cognitive Load
CPS	Cyber-Physical Systems
CPPS	Cyber-Physical Production Systems
HMI	Human-Machine Interaction
IoT	Internet of Things
IT	Information Technology
OS	Operative System
SQL	Structured Query Language

Chapter 1

Introduction

Contents

In today's world, people's interaction with the digital world keeps raising as technology evolves in every field and every interaction with a machine demands some type of controller. Usually an interface, through where the user is able to interact with the machine's functionalities. In a study conducted by Gitte Lindgaard [31], when it comes to Web pages, visual appeal can be assessed within 50 ms, suggesting that web designers have about 50 ms to make a good first impression. In other words, affected on appearance alone, people tend to make quick decisions about what they are seeing so your interface's look will immediately affect how your users feel about it . This applies to any type of software and it is not only about the interface, users tend to quit using applications fairly easy when they are slow or not intuitive. In an environment where the interaction between humans and machines is mandatory for the success of the operation, the attention to detail must not be discarded as it could mean the triumph or failure of a business.

1.1 Context

The Industrial evolution started with the invention of the steam engine, marking the first Industrial Revolution, named "Mechanization". The second one came with the help of electricity and was nominated "Mass Production". Evolving to the use of Electronics and Information Technology (IT), the third revolution was born, titled "Digitization" and today has a result of the use of Cyber-Physical Systems (CPS) and the Internet of Things (IoT) and Services in factories, the forth Industrial Revolution came to live and it is called "Industry 4.0" [23].

The CPS come from the integration between the digital and the physical world and consists of a control unit, usually one or more micro-controllers, which control the sensors and actuators that are necessary to interact with the real world, and processes the data obtained. [23].

One of the core components of the Industry 4.0 are the Cyber-Physical Production Systems (CPPS), in which the CPS concept is applied to the production environment. In order to maximize the control and traceability of every smart component on the factory, each one of these components has a Digital Twin, which is a digital representation of the physical object that allows us to gather information about the component and every sensor and actuator that make it up as well as communicate with other components that belong to the system. With the Industry 4.0 appearance, the human integrates the CPS with its own Digital Twin representation. This way, not only the machines are aware of the existence of other machines but also of humans with which they can and must interact. This integration came from the need to change the roles on the collaboration between both parts, from demanding an adaptation from the human operator to the machine [2], to becoming the machine to adapt to the human.

With the evolution of the manufacturing paradigm, balancing and changing the volume-variety relationship became a necessity. From Mass Production where factories produce large amount of similar products raising the volume and decreasing the variety, to the Mass Customization, increasing the product variety and reducing the volume manufactured of each model [21]. With this evolution, companies are more agile and capable of responding to the client's will which emphasizes the importance of the human intervention on the assembly line, bringing the human's flexibility to adjust to any expected and unexpected changes [15].

The interplay between humans and a CPS occurs either by direct manipulation, or with the help of a mediating user interface and it is best applied in a CPPS, where the machines are responsible for performing the heavier and repetitive operations, while the human employees are responsible for handling shop-floor equipment and supervising processes for high-level decision making [2], aiming this way for the delivery of a high quality product, taking advantages of both human and machine's best attributes to complete a task.

Just like machines, assembly line operators need to be guided during the manufacturing process. An interface that contains every task that has to be performed is used to conduct the operators during the entire process. In each station of the assembly line, a number of tasks, that diverse depending on the product that is being assembled at the time, are displayed in the screen and the operator as a limited time to complete them.

1.2 Problem Statement

There could be an enormous variety of combinations of tasks in each assembly station, which raises the probability of error if the operator is not certain of how and what set of tasks to complete that will fulfill a specific product's requirements, and it can get worse if the factory is capable of producing a large number of different products.

For each product that comes on the assembly line there is a list of tasks to be completed on each station on a short time-lapse (around 228 seconds, for the PSA Group) and when there are multiple products in a row that require the same tasks the operator starts to ignore the instructions, because it is not clear that the tasks changed or the information has bad resolution [48], and simply performs the same tasks that completed on the previous product. Mistakes start to appear when suddenly a different product shows up on the assembly line and the operator was not able to identify the changes on the support interface.

Most of these mistakes, come from the fact that the operator was not capable to detect the changes on the screen between each product that shows up because the interface was developed without taking into consideration the **cognitive load** of the employees, due to the long working shifts, neither their **expertise level**. In most production factories that contain an assembly line and require a human intervention during the process, the working shifts can reach the 16 hours mark and a total of more than 60 hours a week, which is physically and mentally heavy [3, 5] so there is a need to reduce their mental effort when it comes to software that was created to assist the operator. Human operators should not be treated like machines nor expect that they preform like so as they will not perform proficiently as soon as they integrate the production line, there is a learning curve that must be taken into consideration and treated carefully.

Concluding, there is a need to redevelop the support interface system replacing the current one with another capable of adapting to the user's expertise level and an interface that will reduce the operator's mental effort when interacted with so that it really becomes a **Support Interface** instead of just a screen with information.

1.3 Motivation

As the manufacturing capacity of companies continues to grow due to this Mass Customization from the evolution of manufacturing, extending their variety spectrum, this causes an increase of the workload felt by the human element of this production environment [44], as well as a decrease in their situational awareness during the operation. This solution has direct impact on the operators of the company since it is where the problem resettles and indirect change on the company's business.

As for the assembly line operators it is expected that with the resolution of the presented problem they are more attentive throughout the work shift which should lead to a more precise performance which will prompt a higher success rate and consequently boost their confidence and motivation.

With a successful worker comes a successful business. Due to the reduced failure level from the operators, the company is able to produce more in less time and with a higher quality, this will avoid human and material resources waste as the products will more likely get a positive grade when being evaluated by the quality assurance department.

1.4 Objectives

Normally, workload and awareness issues are handled by changing the physical element's (robotic manipulator) autonomy levels, increasing them in order to ease the human operator's intervention. However, we propose that changes are made to the support interface, and to the manner in which it is conveyed to the human operator, providing them with a tool that affects the operator's workload and awareness in a positive fashion [12].

By now, we have identified the two unexplored variables to the development of a support interface for an assembly line worker.

The main objective of this dissertation is to contribute for an easier, more intuitive and adaptable interaction of the production line operator with the interface that provides all the tasks to be executed during the process as follows:

- **Providing the worker with an interface that is able to adapt itself to the user's expertise and reduce cognitive load.**

Over time, this interface will be able to change based on the user's success rate, which is gathered from an evaluation team that makes sure that every task was executed correctly. In order to have an adaptive interface and since this is inserted in a Cyber-Physical Production System, the system needs to know who is working at each shift which denotes to the need of having a virtual representation of each operator, creating a Digital Twin that will communicate with the remaining of the components as well as hold the information about the operator so that it can adapt the interface accordingly.

1.5 Problem Definition

The main problem stated on Section 1.2 relies on the necessity to create an interface that is cognitively light and adapts to the user's expertise. Yet we must define the problem in order to develop a proper solution. Here we will taper the problem statement and end up with a much more specific description of the problem to be solved by defining every influence and disposal variable as well as any general assumption taken.

We have divided the problem into three phases that will help with our development:

1. How will we assess the expertise level of an operator?
2. How will we connect an interface with each level of expertise?
3. How will we create a low cognitive load interface?

Questions number 1 is related with the expertise level side of the problem and question number 3 is focused on the cognitive level aspect as for question number 2, it is divided between both concepts due to how they co-relate and affect each other. From this point forward, we are able to tackle the problem as a pack of smaller problems and this way take short but more accurate steps when aiming for a solution. With these three questions come to mind a number of other questions, some of them might be answered on Chapter 2 and some technical ones that will be answered on the Solution Chapter 4.

For this problem there are a number of variables and assumptions that we must take into consideration when developing and structuring our solution and these are categorized and listed bellow. As **Disposable Variables** we are considering every variable that must not at any point influence how the solution is developed. As **Influence Variables** we are considering every aspect that will affect how any algorithm or method created works. **General Assumptions** are every aspect that we take as guaranteed at the beginning of the assembly process.

- **Disposable Variables:**
  - How tasks are evaluated;
  - Task execution moment during work shift;
  - Operator's physical or psychological disabilities;
  - Screen dimensions;
  - Environmental comfort variability.
- **Influence Variables:**
  - Number of tasks or Time Period analyzed;
  - Department's task complexity;
  - Task execution frequency;
  - Execution time-lapse;
  - Operator's expertise level;
  - Number of Expertise Levels;
  - Evaluation scale;
  - Expertise's information disposed requirements;
  - Cognitive light design principles;
  - Type of information;
  - Information's priority level.
- **General Assumptions**
  - Each operator must have a number/code assigned;
  - Each operator has its own support interface display;

- Every tool and material is previously selected and at the operators disposal;
- Each operator only work in one department;
- All data is already inserted on the Database.

With this decomposition we have accomplished a better definition of our problem. We are looking for a **system** that is capable of **identifying the level of expertise of a user** based on their task completion success rate on a certain time period or number of performed tasks, after the identification, the system must **assign an interface** that goes along with the user needs **depending on the expertise level** assigned. At last, this interface must **follow the design principles** in order to make it as **light as possible cognitively wise**.

1.6 Structure

The present document has the following structure:

1. **Chapter 1 - Introduction:** The reader is presented with an explanation of the context where this dissertation is inserted as well as a brief explanation and a definition of the problem in hand and the motivation and objectives for its solution.

2. **Chapter 2 - State Of The Art:** In this Chapter, we will answer a set of research questions that are crucial to develop the best solution possible for the problem in hands, learning methods and approach that will support our implementation.

3. **Chapter 3 - Concept and Approach:** This third Chapter is used to present the reader with some concepts that are believed to be beneficial for this type of systems and the approach taken to the problem in hand. It is also presented the PSA Group case, that presents the practical case that motivated this dissertation's development.

4. **Chapter 4 - Proposal:** At this point, the reader will be presented with the developed solution and all the aspects that where taken into account during its implementation such as the technologies, the influence variables, the expertise level calculation algorithm and the interface itself.

5. **Chapter 5 - Testing Environment:** The testing environment is described during this Chapter where the architecture, elements, sequence of activities and points of evaluation are described and explained.

6. **Chapter 6 - Results:** In this fifth Chapter there are presented all the results acquired from the tests done to the system and interface.

7. **Chapter 7 - Conclusion:** This chapter wraps up the whole document, analyzes what would be the future work and alternative test that would be interesting as well as an example of where this solution could be applied.

Chapter 2

State of The Art

Contents

Now that we have identified the flaws on the current system, we can seek for existing work and approaches that have tried to solve the problem described on chapter 1. This way we can have a deeper knowledge about new implementation and design plans for our solution proposal as well as it's validation.

This dissertation aims to solve the problem related with the assembly line worker's support interface, focusing on how to develop an interface that could:

1. **Adapt to the user according to their Expertise Level**

2. **Ease the user's Cognitive Load**

In this chapter we are looking to find solutions, concepts and methods for a user interface that is capable to adapt to the user's needs and imperfections, that will support our solution proposal. For so we are looking to answer a set of research questions that can break our issues into smaller questions and this way help us to shape our solution.

1. **Adaptive interface for expertise level?**
2. **How to reduce Cognitive Load?**
3. **How to get the user's attention?**
4. **How to develop a support interface for a CPPS operator?**

Before jumping into answering the research questions we are going to analyze the problem and demystify some concepts attached to it that besides helping us to make a better search and obtain more precise results and support it comes to understand the chosen research questions above.

2.1 Problem Analysis

On this section we will be analyzing the problem in order to make it easier to understand and correlate every question that are addressed to the problem.

2.1.1 Reviewing The Issues

In this section, we will start by retracing the issues previously identified and presented in Chapter 1 by formulating a question for each one of them and explain them and any concept that might not be intuitive. Although the issues are fairly similar, we will analyze them as if we were only dealing with one of them, this way we are able to get a deeper look at each of them and more easily relate them. Let's start by looking at issue number 1 and formulate a question to address it.

Q.1.: How can we create an adaptive interface that adapts to the user according to their expertise level?

Looking at the question as a whole, it might seem somewhat complex as some concepts are not clear and need to be clarified. For **Adaptive Interface** we refer to a screen that should change depending on the user that is interacting with it. This adaptation can be based on multiple factors related to the human (age, height, physical weakness, etc...). The **Expertise Level** side of the questions represents a human variable presented in every Human-Machine Interaction since no one is born taught and there is a learning curve for everything we interact with.

Both of these concepts are investigated and analyzed in Chapter 2, in Section 2.3, where this question is divided into two strong concepts that can work separately and if we were to create two questions out of this one, they could be:

Q.1.1: How to create an adaptive interface?

Q.1.2: How to create and interface for each user's expertise level?

Moving onto issue number 2 and starting once again to formulate a question to represent the issue.

Q.2.: How can we create a low cognitive load interface?

For this second issue we are not going to divide the question into two, instead, we will clarify what we mean by Cognitive Load. As explain on Chapter 1, on production systems, the assembly line workers have really long shifts where they are performing the same task throughout the whole shift and it is expected that after a few hours the operator starts to show signs of tiredness which will be reflected on the success of the tasks. When we are referring to a low Cognitive Load Interface, we are talking about an interface that is easy to understand and where the important information is clearly highlighted so that the worker is able to identify the tasks and any variety to the previous product without much mental effort.

2.1.2 Connecting Both Issues

After analyzing the questions previously elaborated and having a better idea about the problem, a possible solution is now starting to be idealized, but we are still to understand how these two problems relate and only after that we are allowed to work on designing a combatant solution.

We must now gather both questions together, by doing so we will have a more shaped idea of the problem as a whole so that we can envision a proper solution on our head. Let us now read both questions side by side:

Q.1.: How can we create an adaptive interface that adapts to the user according to their expertise level?

Q.2.: How can we create a low cognitive load interface?

Thinking one last time about the problem in hands, by looking to the two questions we realize that we are trying to develop an interface that is both cognitively light and adaptable which means that every time we may try to evolve in one aspect we are indirectly affecting the other which brings an extra challenge to the equation when developing. On the other hand, after all, when trying to adapt an interface to the user, the main goal is to make it so that the user has less effort analyzing the information displayed and so reducing its cognitive load which is what we are looking for when developing a low cognitive load interface.

2.2 Interface for CPPS operator

It is in the assembly line that the Human-Machine Interaction in production systems is presented the most and where companies must focus their preoccupation for a higher success rate and satisfaction from the operators.

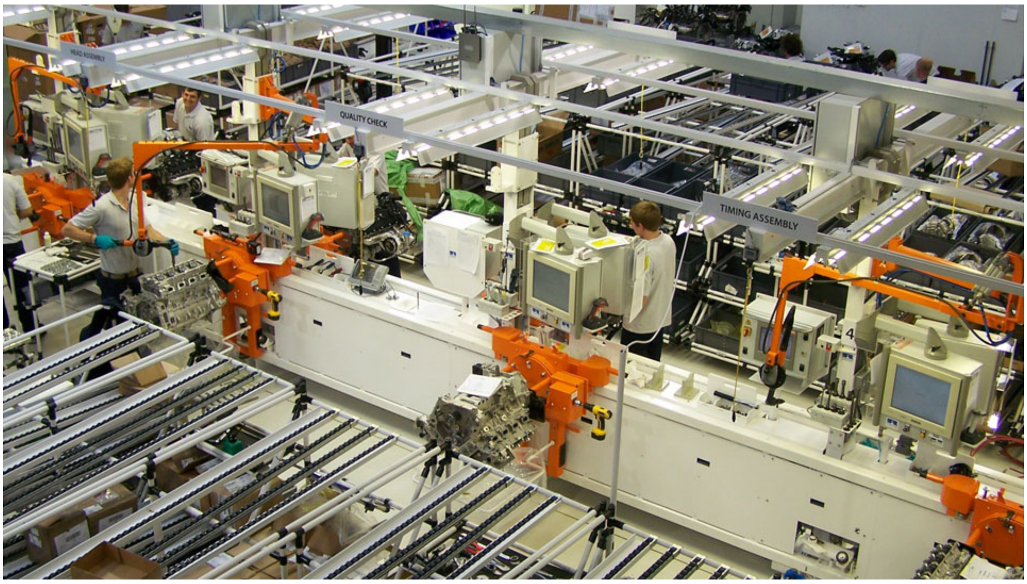
2.2.1 Cyber-Physical Production System

The Forth Industrial Revolution supports itself on the dream of combining both real and virtual worlds by connecting every physical object to each other and enabling information sharing between all of the elements of the system and this way increase the flexibility and visibility for data handling and access in production systems [46]. With this flexibility comes a whole new concept to production by increasing the factory's spectrum of products variety, responding faster to the market necessities and trends by developing custom products in low quantities avoiding resources and time wastage.

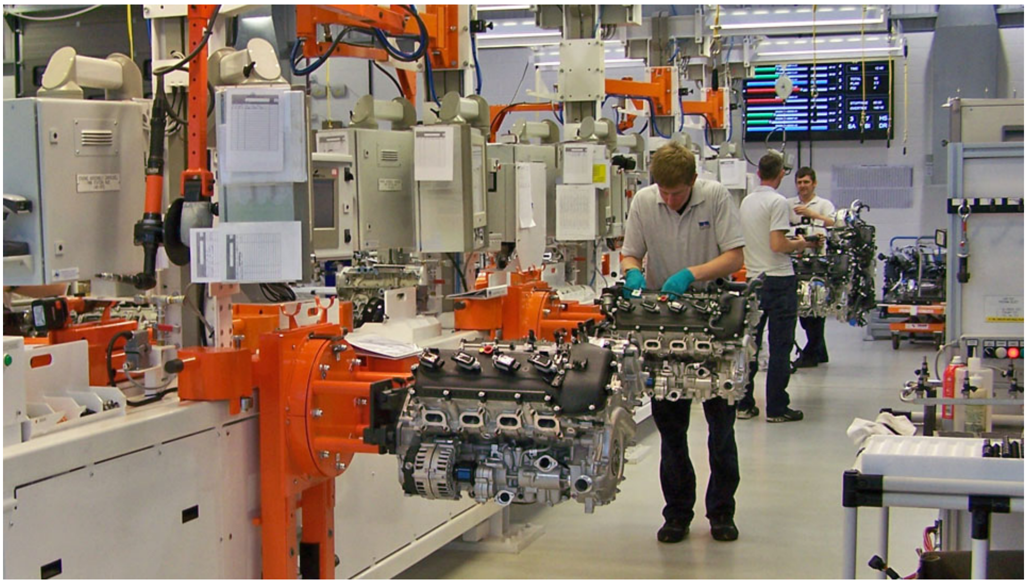
In order to improve the cooperation of field devices and industrial plant personnel there is a need to consider every actor (e.g. computers, sensors, production machines, robot's and humans) in a production line as a manufacturing component. Once an actor in the production line is digitally abstracted (Digital Twin) by the factory, it becomes a smart component. This abstraction provides a physical component with smart capabilities, which enables it to become an active part in the Smart Factory (CPPS), and consequently in the production environment [40].

2.2.2 The Interface

As explained on the Introduction's sections Context 1.1 and Motivation 1.3 the production line operator as a limited time to perform a number of task that may displayed on a screen or in a sheet of paper. With this screen or paper, the worker must be able to identify the task and every tool that is demanded to successfully complete the task as fast as possible so that he can be focused on the execution rather than reading the instructions. With the Mass Customization manufacturing paradigm the margin of error shortens which enhances the importance of a well developed interface where all the topics evaluated on this chapter must be taken into consideration.



(a) Assembly Line with Multiple screens



(b) Assembly Line progress screen

Figure 2.1: Assembly Line interface example [37]

Ricardo's Technical Center in Shoreham is an engine production company that produces 4000 engines annually, represented in the images above (Figure 4.7). It is composed of a number of stations for the production of a high quality engine and each station is equipped with a MacDonald Humfrey 'Human Machine Interface' (HMI) providing guidance to each operator on the precise sequence of operations required at each stage of assembly (2.1a). The tools and assembly equipment used at each station are instrumented, and provide data directly into a central warranty database for each engine built, providing complete finished product traceability (2.1b) [37].

2.3 Adaptive Interface for Expertise Level

Interfaces can be adapted to users in a variety of aspects and variables, but for this dissertation, we are focusing on the expertise level and how to adapt the interface accordingly. For such we need to understand how can a system be able to adapt to the user and what are the user's needs on each level of expertise.

2.3.1 Adaptive Interface

The search for an interface that could adapt to each user and offer an enhanced user experience, referred to as Adaptive User Interface (AUI), has been, for a long time, a hard challenge for software developers and computer science researchers [34].

Briefly, an Adaptive Interface can be defined as an interactive software system that improves its ability to interact with a user based on partial experience with that same user [30].

There are three main ways where this research area can go [34]:

1. **User customization**
2. **Explicit request for input from the user**
3. **Machine Learning and data mining techniques**

2.3.1.1 User customization

With this approach, the user is aware of the adjustments made by the interface, where he is given the possibility to choose from a spectrum of interfaces and select the one that better satisfies its will or change the position of some of the components that compose the interface, passing this responsibility to the user. This type of adaptability is present in the majority of software such as internet browsers (Google Chrome, Mozilla Firefox) or text and code development editors, where we are able to change the theme, colors, font type, etc... and one particular example is the Operative Systems (OS), for instance, on the iPhone, with the Apple's OS or on Android phones, with the Android OS, where we can drag and organize any app to whatever position better satisfies our personal interests, making them more accessible.

2.3.1.2 Explicit request for input from the user

Similarly, to the previous approach, here the user is also aware of the changes due to the fact that the user inserts its topics of interest and the system suggests content related to the selected topics. The system is not learning with the user's interaction, the user is explicitly telling the systems exactly what he is looking for and the interface adapts showing the desired content and other material related with the ones the user has seen.

2.3.1.3 Machine Learning and data mining techniques

Unlike the other two solutions, the Machine Learning and data mining techniques aims to provide the best interface to the user. Extracting information from the user's interaction with the interface, recognizing patterns and frequent activities, changing the page content and giving suggestions based on what the user has searched without him being voluntarily a part of the process. This type of interaction is visible in websites like Amazon, Netflix, YouTube that recommend movies, videos or products for the logged user based on his logged history. A similar approach is the "Smart menus" feature that came with the Windows 2000 where the idea is to place the most selected options on an easier access point in the menu, and the not so frequently accessed sub-menus are hidden and will only return to an easier positions if the user selects them with enough frequency so that they can be qualified as frequent choices and so come back to a more desired position in the menu [34].

2.3.2 Expertise Level

Expertise can be defined and divided into several groups and it all depends on the project in hand. The only thing that is present in every expertise level collection is the concept of **Novice**, **Intermediate** and **Expert** as they represent the core and margins of the categorization spectrum.

Every user starts as a novice and makes his way all the way to become a product expert going through all the phases in between. This evolution will depend on multiple factors such as **how much time the user spends interacting with the interface** or **how familiar is he with technology** and others [51]. Having this in mind we need to also understand what does a Novice need to easily understand an interface as well as to improve to become an Expert and also understand how an Expert will use the interface and how to make it as user-friendly as possible for its purpose.

Table 2.1: Design principle and expected behaviors by Expertise Level [51, 20, 36]

Level	Design Principle
Novice	Linear task flow Allow to work with very limited domain knowledge Step by step work flow Simple and Clean Interface Easily discoverable help text
Intermediate	Don't need scope and purpose explained Usually establish the functions they use with regularity May still be uncertain about some tasks
Expert	Know their tasks and how to complete them Provide maximum flexibility in determining task flow Provide shortcut keys

The above Table 2.1 describes some design principles and expected behavior for the three major expertise levels (**Novice**, **Intermediate**, **Expert**) to have in mind when developing any type of interface.

When first engaging with an interface (Novice), users should have the ability to go through a set of tasks and interactions with few or zero domain knowledge which comes to advise to develop a linear task flow with few decision points. As the users become more competent (Intermediate) they start looking for ways to speed up the workflow by identifying shortcuts or finding patterns on the system's feedback, but still are uncertain about some tasks. The highly competent users (Experts) are capable of completing any task with minimum cognitive effort as they understand the system and know exactly every interaction and reaction and have identified every pattern that the system may have.

It is clear that the user's requirements for each expertise level are highly scattered and each one can have vastly different tactics for being successful with the same software, yet not every business has enough money and time to develop multiple interfaces for each level of expertise. It is most common to direct the attention to the novices as everyone starts as a novice, however, studies show that the majority of the users are neither novices nor experts but intermediates as identified by the bell curve present on Figure 2.2 [32, 7].

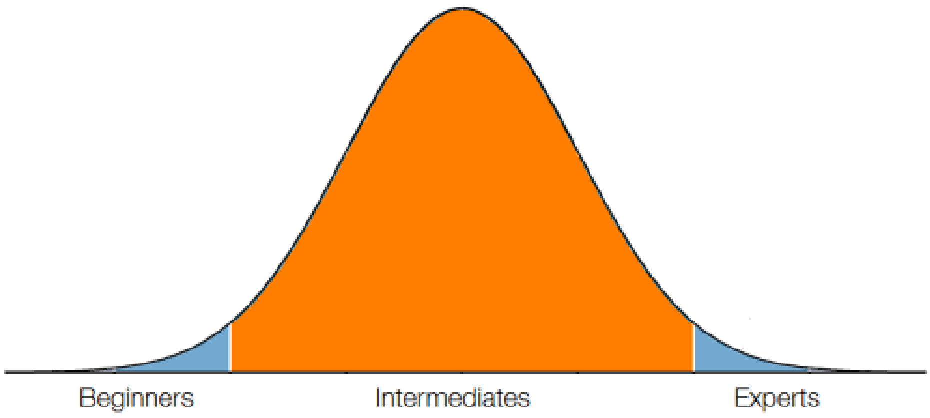


Figure 2.2: Activity flow on products by expertise level [32]

An example of a web page that is suitable for all users along the spectrum, from novice to expert, is the Google's homepage (Figure 2.3). It has a fairly simple and clear interface. First time users are encountered with a pleasing and inviting logo on a wide light colored background with a large text box under it that almost speaks for itself and two buttons under it that are clear about their action. As for expert users, they are presented with the exact same interface and supported by all the light gray items inserted in the page's frame and top right corner of the page that give them access to other features such as Setting or Gmail [51].

To novice users these items are ignored as they blend into the background, on the other hand, expert users find these to be extremely useful as they help to accelerate the work with less frustration [51].

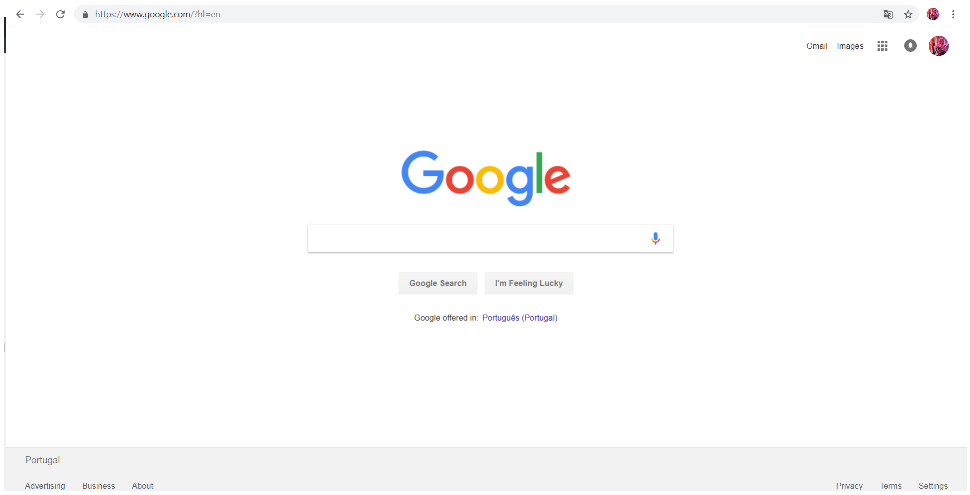


Figure 2.3: Google's Homepage [51]

2.4 Cognitive Load

Cognitive load is described as being the total amount of mental effort that is required to complete a task involving processing of information [43] which can easily be confused with cognitive overload that occurs when the volume of information supplied exceeds the information processing capacity (working memory) of the individual [14, 25]. Cognitive load can be divided into three groups (The Good, the Bad and the Ugly) as represented in Figure 2.4

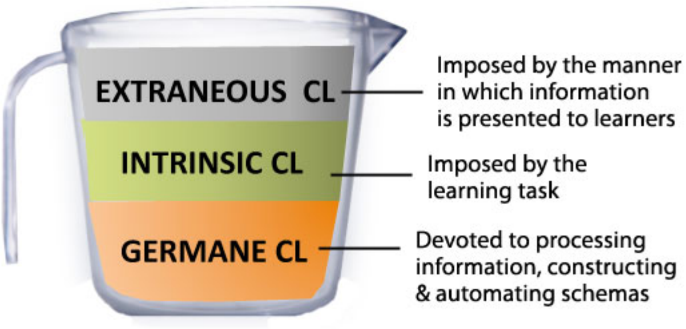


Figure 2.4: Three types of Cognitive Load [35].

1. **Germane CL**: Refers to information that helps the learner process the new information
2. **Intrinsic CL**: Concerned with the natural complexity of information that must be understood and material that must be learned and it is stationary and cannot be changed other than changing the basic task or knowledge levels [49].
3. **Extraneous CL**: When material is presented in a way that it does not build schemata or interferes with any schemata construction [45].

Cognitive load is reduced when material is presented in the right form, as John Sweller, the emeritus professor of University of New South Wales and the educational psychologist who has created the cognitive load theory in the 1980s, explained in a short talk with a simple example [43]:

> *"Lets assume I show you a 10cm by 10cm square, it will take you a fraction of a second to recognize that is a square, you know is a square as soon as you look at it. That's presented in visual form. I could present a square in spoken form (...) for example, think of an object that consists of a line 10cm long which is vertical, think of another line at 90º to the first line where the ends of he two lines meet and assume that it is also 10cm long and it is horizontal, thing of a third line joined to the second line and parallel to the first line at 90º to the second line, 10cm long (...). This sort of information should never be presented in verbal form, it should be presented in spacial 2 dimensional form (...)."*

John Sweller sums up his short talk referring that some materials naturally lend themselves to be presented in spacial form as other materials lend to be presented in verbal form.

2.4.1 Reduce Cognitive Load

As identified previously, it is fairly easy to get overwhelmed by a deluge of complex information as working memory has a limited capacity for processing information. Yet there are a few ways to reduce this cognitive load and prevent the users from having a bad experience with our interface.

2.4.1.1 Understand how users read online

According to the famous eye-tracking research by Norman Nielsen Group, the way people move their eyes when reading an online document creates an F-shaped pattern and it happens because the user starts by reading in a horizontal movement across the upper part of the content area, then the user moves down the paper and reads on a second but shorter horizontal movement and finally the user scans the remaining content on the left side in a vertical movement as it is demonstrated on the Figure 2.5 below [41].

Figure 2.5: F-Pattern by Jakob Nielsen [41]

Having this in mind, it is suggested that we reduce the number of columns to only one and second split the body text with horizontal lines with bold lines to make it easier to scan the text.

2.4.1.2 Use schemes that already exist and users know them

There are some actions, interactions and symbols/icons that do not need to be reinvented since we are acquainted to them and use them every day. Such things as recognizing a certain object as a representative of a category of objects [43].

Therefore we should stick to the conventional definitions which will make it easier for our users to go through an interface, avoiding unnecessary extra time to learn any new interaction and instantly recognize and execute the desired action [43].

2.4.1.3 Make instructions simple

Related with the previous topic, good design should be self-explanatory where the communication between the product and the user is immediately identified. Taking the Google search page as an example, the user easily recognizes what the main purpose of the page is and besides that, if we analyze it with a little more detail, it is noticeable that the cursor automatically starts on the search box, avoiding the user from having to grab the mouse and click on the search box, allowing an immediate search by simply start typing [43, 18].

2.4.1.4 Get rid of everything that is not essential

Keeping the interface nice and clean is one of the most important things to keep your users focused on what's important in your page, reducing loading times and streamlining the experience. This is where we understand that less is more and according to a study on how aesthetics affect a website's first impression [50] acknowledged, most users appreciate a more simple and clean website over a complex one [18] and that is why we must only use relevant graphics and only display the ones we have to. It is also very important to diversify on the visual elements, balancing the amount of text, images, videos and others on our interface.

2.4.1.5 Keep a consistent format throughout

Pattern recognition has been considered by many, the essence of human behavior [33] and that is what we look for in everything that we as humans try to learn in order to have an organized structure of the world around us and that is why when building an interface, it must be coherent and follow a logical pattern. We tend to interact better with an interface when it is organized when it comes to colors, text size and typography, labels structure and descriptions, everything must be in sync.

A great example of a website that is both visual and functional consistent is Pinterest. Here the format remains the same throughout the whole page, making it extremely easy to use.

2.5 Get User's Attention

Information should be structured to capture user's attention. Attract users to engage with a web page is one of the most challenging yet most important aspects online business are facing nowadays.

Every system or web page has a target audience for which the page must be developed for and it is crucial that we establish a good description of our audience yet there are some facts about humans that must always be taken into account when developing any type of interface and this way get a step closer to its successful interaction [24]:

- We seek and use visual structure
- Our color vision is limited
- We perceive what we expect
- Our vision is optimized to see structure
- Reading is unnatural
- Our peripheral vision is poor
- Our attention is limited; Our Memory is imperfect
- Recognition is easy; Recall is hard

In his blog post, titled *Best Ways to Attract and Hold Online Users' Attention, Part 1 (2018)*, Kucheriavy, A. informs that records say that in 2000 the average human attention span was about 12 seconds, by 2013 it decreased to 8 seconds and it has continued to fall since and states that people can only pay attention and deal with a single piece of content at once [27]. These numbers reflect the information overload that users are attacked with and its processing has become an annoyance and time-consuming as users dislike spending time and cognitive ability analyzing and understanding irrelevant information. In order to make the most of these few seconds our interface's first engagement with the users must be the attractive to keep the user on our page. There is a manifoldness of techniques and tips on how to grab our audience's attention, for instance, using emotions, physical needs, self-made choices but the two most effective ones are **Contrast** and **Novelty**.

2.5.1 Contrast

Contrast is everywhere and it is what gives us the ability to intensify differences which enables designers to define priorities and properties of an element over the others [52]. The phenomenon was introduced by Helmholtz in 1866 after understanding the size effect observed while a sports announcer interviewed a basketball player and a horse jockey [9].

There are multiple ways to create contrast on a page using **Colors**, **Size** and **Depth** by grabbing and joining antonymous word to create attraction:

- Closed and Open
- Full and Empty
- Big and Small
- Filled and Outlined
- Ordered and Chaotic
- Textured and Flat
- Letter spacing and No Spacing
- Square and Round
- Filled and Transparent
- Rough and Smooth
- Horizontal and Vertical
- Old and New
- Strange and Familiar
- Lined and Dotted
- Detailed and Clean

Through these three elements, we are able to create a great contrast that leave our users attached to our page.

Color is the first topic that most people think about when referring to contrast. In fact, it is the easier way to distinguish what is important from what is **very important**, purely by changing the color of the text or with a simple **bold** word. Figure 2.6 shows an example where through color we get attracted to different things. All colors can be grouped based on their temperature: warm, cool or neutral. Every color that remind us of the heat and fire belong to the warm group (red, orange, yellow), the cool group enhance the water and the cold (blue, green, purple), the neutral group is composed of the black, grey, white and brown. Mixing color temperatures can create a dramatic contrast specially when merging warm and cool colors [26].

Figure 2.6: Color Contrast example [42]

SIZE is only evaluated when compared to something else, we describe a 180cm tall sports announcer has short when he is interviewing a basketball player, but such thing does not happen if he is interviewing an artistic gymnast where the announcer will be considered as tall, the apparent size of the sports announcer is affected by size contrast, which exaggerates his relative tallness or shortness against the surrounding context of athletes. Thus, Figure 2.7 shows the Ebbinghaus illusion (frequently called Titchner's circles), in which the central circle surrounded by large inducing elements appears smaller than the central circle surrounded by the small elements [9]. This concept is applied on the "Big and Small" example listed above, where we place BIG text next to small text to create this size illusion and drive the user's attention to where the most important information is as demonstrated in Figure 2.8.

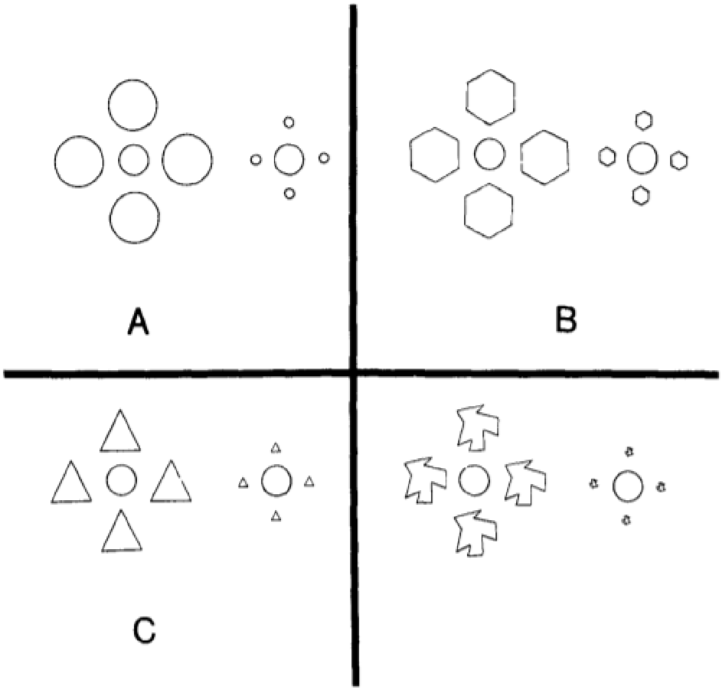


Figure 2.7: Four variants of the Ebbinghaus illusion [9]

Figure 2.8: Size Contrast example [47]

Depth creates the illusion that the page's content is closer to the user and popping out of the screen (Figure 2.9). As Humans, we live in a 3 dimensions (3D) world and objects around us exist in 3D, but on the screen there are only 2 dimensions which opens up a new window of opportunity to create contrast, by adding the missing dimensions we are used to adding back a sense of reality.

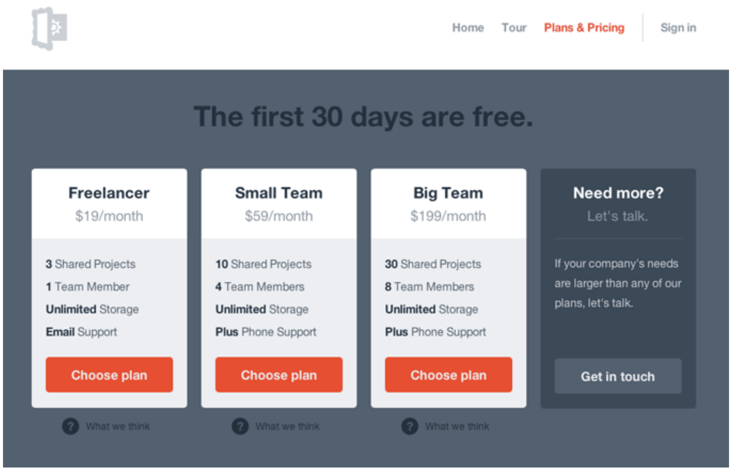


Figure 2.9: Depth Contrast example [19]

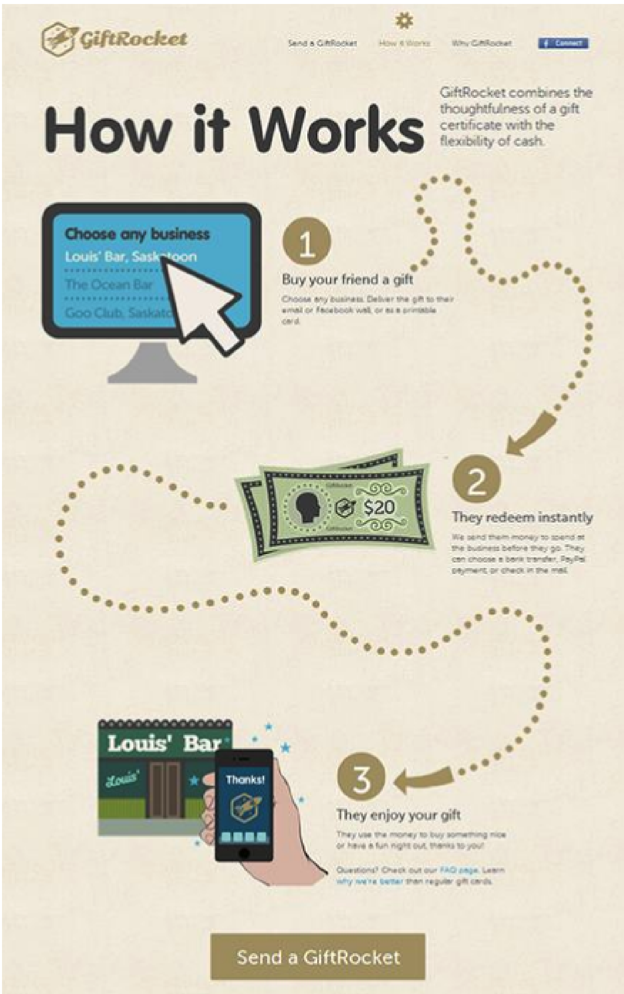
It can be achieved using the two elements mentioned before (Color and Size), light and shadows and here are some of the different cues that can be used to convey the illusion of depth:

- **Occlusion (overlapping objects):** When one object obscures part of another object. Nearer objects occlude (cover up) objects that are further away [4].
- **Size and scale:** Larger objects appear closer and smaller objects appear further away [4].
- **Cast shadows:** A shadow cast by one element on another gives cues about their relative distance. Drop shadows are perhaps the most common way one adds depth [4].
- **Lighting and shading:** Much as light adds depth by casting external shadows, it also shows depth in how it acts over the surface of one object. How light plays over a surface gives clear cues about the shape of that surface in all 3 dimensions [4].
- **Depth of field (focus):** The eye focuses on one object and as it does other objects become slightly blurry, while the focused object remains sharp [4].
- **Color:** Cool, dark colors tend to recede into the background. Warm, bright colors tend to move to the forefront [4].

2.5.2 Novelty

Novelty is a concept that refers to the use of new, different or unusual elements and anything that is new captures our attention and motivates us weather it is a new phone, new clothes, new working environment and that is the reason it is used on interfaces. In this case, novelty comes to avoid text-heavy pages by inserting images, videos, facts, graph that will trigger this impulse on the audience [8, 28, 29].

(a) Webpage before novelty [29]



(b) Webpage after novelty [29]

Figure 2.10: Novelty application example

Figure 2.10 is an example of how using novelty can clearly boost an interface to become much more attractive and engaging. On the top image (2.10a) the page does not draw any attention when looked at where the only thing that contrasts is the title. Now on the second one (2.10b) there are a lot of elements such as images, big numbers and a dotted line that make the users get stuck to the page and wanting to read more and assimilate the information displayed.

2.6 Summary

Each of the research question gets us a step closer to an ideal solution, contributing to creating an interface able to answer both of the variables identified as the main worker's problem.

Solutions for Question One (Section 2.3) led us to comprehend that there are several ways to adapt an interface to the user and how to implement any of them and understand expected behaviors and interactions of users with an interface according to their expertise level. Question number two aims to directly combat the cognitive level variable and the answer in Section 2.4 is giving us different techniques on how to develop an interface that will not overload the user with information. The third Question (Section 2.5) enables us to find ways to guide the user to the critical and relevant information presented at the time and this will vary depending on the user's expertise level. Finally, responding to the last Question (Section 2.2), we are aware of how the system works and how to incorporate the desired interface to this type of business.

Chapter 3

Concept and Approach

Contents

3.1 Introduction

In this Chapter there are presented some believed good topics about the approach to take in this type of systems giving a slight idea of how the actual solution, addressed on the next Chapter, was developed as well as the concept of the thought solution and in which industrial environment it was inspired.

When working with humans it is always highly challenging to satisfy everybody involved regardless the topic as every one is different from one another in many different aspect, either physically or psychologically and that is why one solution will never fulfill everyone's needs. This diversity is what makes it very important to identify in which aspects the solution is looking to work on by defining a constant value for every other possible variable that could be affected by the solution and this way restrict the group of users it is aiming to satisfy and test.

3.2 PSA Production Unit

The PSA Group is a French automotive industry with over 200 years that produces cars for multiple brands [16]. With multiple factories spread along France, Spain and Portugal, the Portuguese one is located in Mangualde, with one thousand contributors, and there they are responsible for the production and assembly of the light commercial vehicles of the group such as the Peugeot Partner and Rifter, Citroen Berlingo and Berlingo Van and Opel Combo [17].

This dissertation is integrated on a larger project that aims to solve the problem of high cognitive load on the factories workers as a way to try to reduce the number of defects detected at the end of the production line and solved in a Quality section which is the last one before the vehicle heads to the client and if the operator on this section does not amend the defects properly, the vehicle will be shipped to the client with defects.

The PSA group of Mangualde decided to do a 45 days study to three different shifts as a way to know what was the average number of defects per car at the end of the production line and here are the results:

Table 3.1: Results of the 45-day study done to detect the average number of defects per car

	Shift A	**Shift B**	**Shift C**
Produced Cars	2370	3185	2532
Average cars per day	52	70	56
Total number of Defects	6867	10733	6322
Average number of Defects per car	3	3	2

The results presented by the table above show that every single car produced has an average of three defects when produced which tells us that 100% of the cars have defects meaning that the Quality station has to rectify every produced car and these numbers are the main motivation for this dissertation as it looks to help to reduce them as much as possible because when translated to working hours and cost these results play a significant role on the business success.

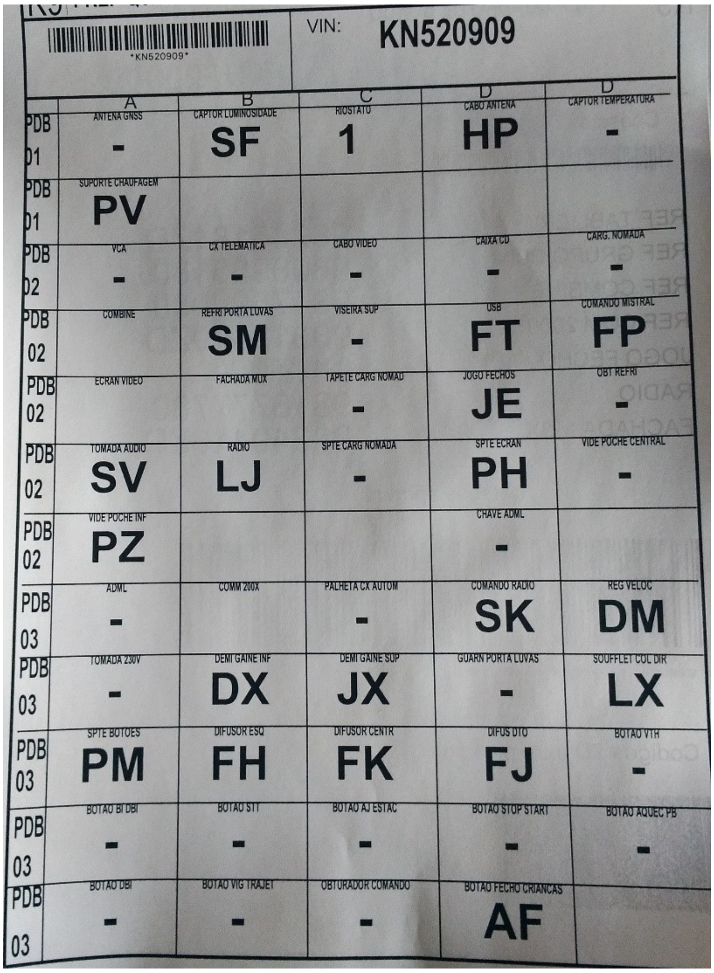
3.3 Manufacturing Flow

Every produced vehicle goes through four manufacturing stages before being ready to deliver to the costumer. It all starts at the **Hardware** line that worries about assembling the core structure of the car by welding all the parts together. It then moves to the **Painting** stage where they will make sure that the car will be able to resist to the exterior demands, not only aesthetically but also sealing wise. The third stage is the **Assembly** line which is divided into several phases, it is here that all of the components, around 2050 pieces and 600 bolts and nuts, are attached to the car, from wiring to engine and seats. The vehicle then moves to the final stage of the production, the **Quality** stage, where 100% of the car is verified for its looks, accordance, effort and noises. Apart from these four stages, there is another one, named **Logistics**, that is responsible for delivering all of the necessary tools to the other four stages with as much quality and less cost as possible[17].

The proposed solution, presented later on the document, was developed and design for the workers on the Assembly line, due to the fact that it is where most of the car is assembled and there is a higher chance of mistakes as there is a really high amount of pieces used, yet it could easily be used by the Quality stage workers and with a few adjustments it would be possible to integrate the solution on all the other stages of the manufacturing process.

3.3.1 The interface

As of right now, the interface for the assembly line workers is non-existence. The operators know which tasks to execute and assets to use through a sheet of paper, named FAV, shown on Figure 3.1, that is printed at the beginning of the line and follows the vehicle along the assembly process. The FAV, contains information, in form of code, about the assets necessary to assemble the car in each station.


KN520909

VIN: KN520909

	A	B	C	D	D
PDB 01	ANTENA GNSS -	CAPTOR LUMINOSIDADE SF	RIOSTATO 1	CABO ANTENA HP	CAPTOR TEMPERATURA -
PDB 01	SUPORTE CHAUFAGEM PV				
PDB 02	VCA -	CX TELEMATICA -	CABO VIDEO -	CAIXA CD -	CARG. NOMADA -
PDB 02	COMBINE	REFRI PORTA LUVAS SM	VISEIRA SUP -	USB FT	COMANDO MISTRAL FP
PDB 02	ECRAN VIDEO	FACHADA MUX	TAPETE CARG NOMAD -	JOGO FECHOS JE	OBT REFRI -
PDB 02	TOMADA AUDIO SV	RADIO LJ	SPTE CARG NOMADA -	SPTE ECRAN PH	VIDE POCHE CENTRAL -
PDB 02	VIDE POCHE INF PZ			CHAVE ADML -	
PDB 03	ADML -	COMM 200X	PALHETA CX AUTOM -	COMANDO RADIO SK	REG VELOC DM
PDB 03	TOMADA 230V -	DEMI GAINE INF DX	DEMI GAINE SUP JX	GUARN PORTA LUVAS -	SOUFFLET COL DIR LX
PDB 03	SPTE BOTOES PM	DIFUSOR ESQ FH	DIFUSOR CENTR FK	DIFUS DTO FJ	BOTAO VTH -
PDB 03	BOTAO BI DBI -	BOTAO STT -	BOTAO AJ ESTAC -	BOTAO STOP START -	BOTAO AQUEC PB -
PDB 03	BOTAO DBI -	BOTAO VIG TRAJET -	OBTURADOR COMANDO -	BOTAO FECHO CRIANCAS AF	

Figure 3.1: FAV

The sheet, as it is visible on Figure 3.1 is divided by stations, listed on the left side column of the paper and the needed assets on the lines correspondent to the station column. Taking an example, when looking to the PDB01 station, the assets used by this station are described on the first two lines. If a box has a dash, it means that the product being assembled does not take that piece, on the other hand, the ones with the code lets the operator know that the product needs that piece.

Although the necessary pieces are already separated for each vehicle which ease the operator's effort, it is still necessary to have a second control on the station by bar scanning each piece that gets placed into the vehicle to make sure that there were no errors of picking.

The only exception is the Quality station located at the end of the line where there is a screen with a list of defects that the car has and need to be repaired as Figure 3.2 demonstrates.

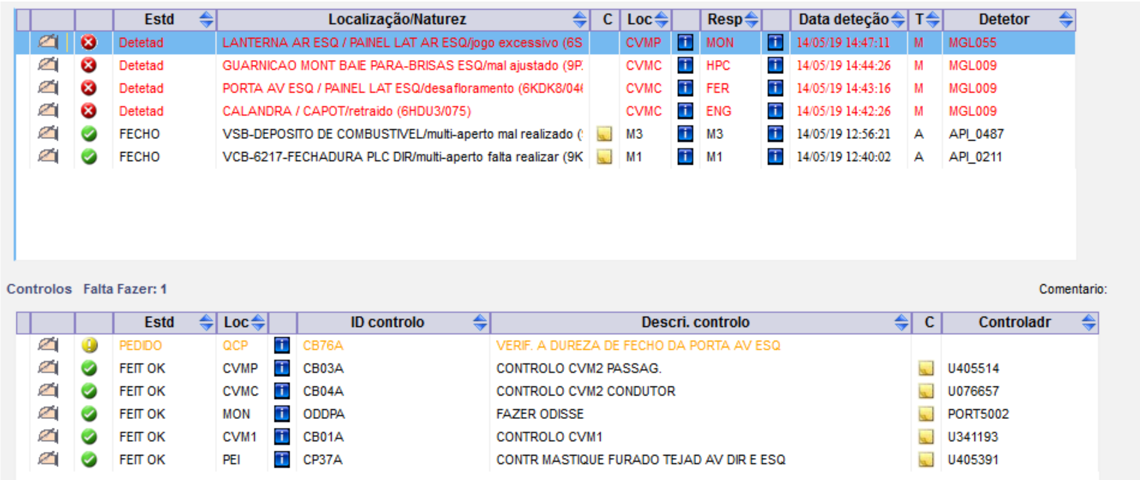


Figure 3.2: Screen shown at the Quality station

This screen is divided into two. The top part is a list of all the defects found in the vehicle, in red the ones that have not been rectified yet and in green the corrected ones. The bottom part of the screen presents the controls made to the vehicle along the line, in yellow the missing controls and in green the accomplished controls. At this station the tasks demand really high levels of concentration by the worker as this is the last station before the vehicle goes out to the client and the way this screen is showing the defects makes the worker waste to much precious time on reading and analyzing the defects as they can be of any nature since this station has to be capable of executing any task from any other station on the line.

There are several reasons that make the operator waste a lot of time when working due to the number of defects that get to this stage of the manufacturing process. The worker is presented with a list of defects that can be of any nature and depending on it, the worker starts the analysis and proceeds to the correction of the defect, when rectified, the worker has to get back to the screen and manually confirm the correction of the defect. This problem is not to critical when the vehicle has few defects, yet if the vehicle being rectified has a vast number of defects, the worker might not be able to address every defect or has to go back to the screen to recheck the information or forgot an element of the list. All of these situations entail time waste and could be crucial to the successfully resolution of the defects which might be the difference from a car being considered ready for commercialization or having to be rectified all over again.

3.4 Approach

With the presented problem it is already known and filtered a group of users that the solution must satisfy, yet in within this group of users named assembly line workers, which could have ten or one thousand people, each one has its own strengths, weaknesses and body structure.

As identified in the Problem Statement section on Chapter 1 this solution look to help the assembly line workers by reducing their cognitive load and also take into account their expertise level by developing an easier way for these workers to know what are the tasks that they have to do and what assets to use for a specific task. With this problem in mind, and being aware that besides the number of tasks and it's diversity, the complexity of each task also varies hindering the worker's job and their workload, it was decided that a good way to address this problem was by transferring from a paper perspective to a flexible and adaptive digital solution by designing and developing an interface that would display the tasks in a way that the worker could easily identify everything that is needed without missing out on any task nor asset.

3.5 Smart Component

Industry 4.0 brought the ability to use CPSs by providing an easier and more intelligent communication between the components of the system, so called smart components. These smart components are non other than machines connected to the system with a digital abstraction that enables them to communicate with each other as well as deliver data about themselves to a monitoring system controlled by the Human making it easier to be aware of any problem happening with the overall system. This abstraction provides a physical component with smart capabilities as it becomes part of the production [40, 46].

Chapter 4

Implementation of the proposed solution

Contents

4.1 Introduction

With this chapter we are looking to understand how the proposed solution was developed by covering all of the aspects that integrate it including the technologies used, the variables values defined, the expertise level calculation algorithm's logic as well as some interface previews. The presented solution was structured in order to be able to adapt to different environments with the minimum effort possible, not being restricted to the proposed nor tested environments. This way, the tested scenario tries to replicate a factory assembly line which is explained later on this document.

4.2 CPS

The Support interface implementation and all of the algorithms related, were developed towards the construction of a larger system containing three major components, the **Support Interface**, a **Robotic Arm** and an **Artificial Vision System**, used to later test the proposed solution as a part of a smaller version of an assembly environment.

4.2.1 Elements

These are the elements that compose the constructed CPS:

- **Worker** - Person that performs the tasks;
- **Support interface** - Contains the tasks to be performed;
- **Robotic Arm** - Passes out a box of Assets;
- **Vision Camera** - Analyze the tasks executed;
- **Box of Assets** - Box containing everything that is needed to complete the tasks;
- **Table Area** - Table divided into 4 Areas;

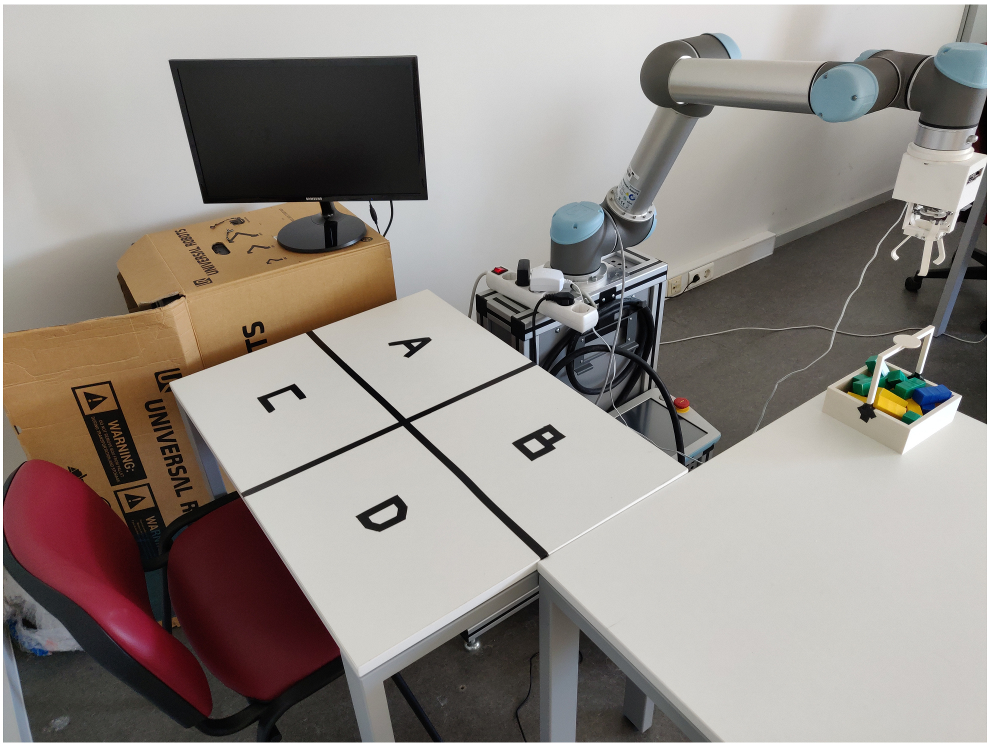


Figure 4.1: Testing area with some of the elements

Apart from the worker and the Vision Camera, all of the elements described above can be seen on Figure 4.1, making it easier to visualize the testing area as it was used. The Visions Camera is not visible on the picture due to its position being in the ceiling, looking down to the working area in a perpendicular direction. Figure 4.1 represents all the elements on a pretesting stage, the box with the necessary assets, situated on the right-hand side of the frame is positioned at its pick up location, the robotic arm's griper, shown above the box, is in it's starting point with the claw open, the Support Interface is presented on the monitor, seen on the top left side of the image, on top of the card box and finally the working area is easily detected as the table, divided by the 4 areas (A, B, C, D).

4.2.2 Description

The test case aims to simulate one Station of the production/assembly line from the moment a worker starts the working shift until the evaluation moment.

In this process, the worker must execute all the tasks presented on the **Interface** by placing the assets on the different areas defined within the **Time Period** previously defined and shown on the interface in a countdown format. The process starts when the user enters the working station and enters his user code, after this, the **Interface** receives the Product being assembled at the time and gets all the tasks associated with it from the database, at the same time, the **Robotic Arm** grabs the **Box** with the Assets and gives it to the user. At this point, the user has gathered all the conditions to start executing the tasks. When the time ends, the **Vision Camera** analyze the **Table**

Area and verifies the final result. In case of emergency, i.e the time is ending and the user feels that he won't be able to finish the tasks, the user has the ability to press the **STOP** button which will stop the whole system including the **Robotic Arm** and **Timer**.

4.3 Technology

In this section the reader is presented with the different technologies from which the developed system is composed by. Each subsection gives a brief explanation about the technology and a small justification of its use for this project.

4.3.1 JavaScript

JavaScript is a scripting language mainly used to develop Web pages but it is not restricted to it, also being used in many non-browser environment such as the presented one. It is used as a client-side language, meaning that it is used to program and design how a page or interface behaves and reacts to any user interaction [39, 38]. JavaScript comes as a first choice for this project for its lightweight, easiness to learn and great community and support for developers.

4.3.2 Electron

Electron is a framework that enables developer to create a cross platform desktop app using Web technologies such as HTML, CSS and JavaScript, having been used to develop fairly well-known applications like Skype, Slack, Visual Studio code, Atom and much more [10]. By also being an open source framework, maintained by GitHub and active community of contributors, it is constantly being improved and its latest update and stable release, at the time, is Electron 4.1.4 released on April 04, 2019 [10]. Since this is an interface to be available at a factory floor, the option to make it a desktop app to be installed in any type of device from a tablet to a desktop computer felt like the right path to take.

4.3.3 Atom

Atom is a cross platform desktop text and source-code editor application that uses the Electron framework (presented on section 4.3.2). It is open source and available on GitHub for possible improvements by the community. Atom comes with a series of packages and pre-installed themes that enables users to customize and style their own editor [13]. For this project, the choice of using the Atom was based on its popularity and previous positive experience using it.

4.3.4 MariaDB

MariaDB is one of the most popular database servers in the world and it was developed by the original MySQL team. It is used by some of the most popular websites on the web such as Wikipedia and Google. MariaDB is developed as an open source software and as a relational database as it provides an SQL interface for accessing data [11]. As for the present project, the choice regarding using MariaDB was based on previous use of relational databases and by having a great community and online support.

4.4 Variables

As explained in Section 1.5 in Chapter 1, there are multiple variables that can influence the algorithm's outcome, and so, deliver different expertise level for a certain an evaluated worker. This way and in order to be able to test the system, a value was applied to some of the previously described variables based on some factors including the lab's available technologies and space, as well as some research demonstrated on Chapter 2.

Some of the variables identified earlier are analyzed on the next Chapter related with the testing phase as they also depend on the time available to test the system. Although, at this stage, we can define a few of the influence variables:

- Number of Expertise Levels: **3** (1 - Beginner, 2 - Intermediate, 3 - Expert);

- Evaluation Scale: **1-3** (1 - Incomplete, 2 - Completed with flaws, 3 - Completed);

- Department's task complexity levels: **1** (Only one level of complexity);

4.5 Implementation Model

For this project, a relational database was constructed in order to store every component that composes the developed system. It was developed following an assembly line's structure and logic making it flexible to be applied in other environments other than the tested one, explained on the next Section 5.

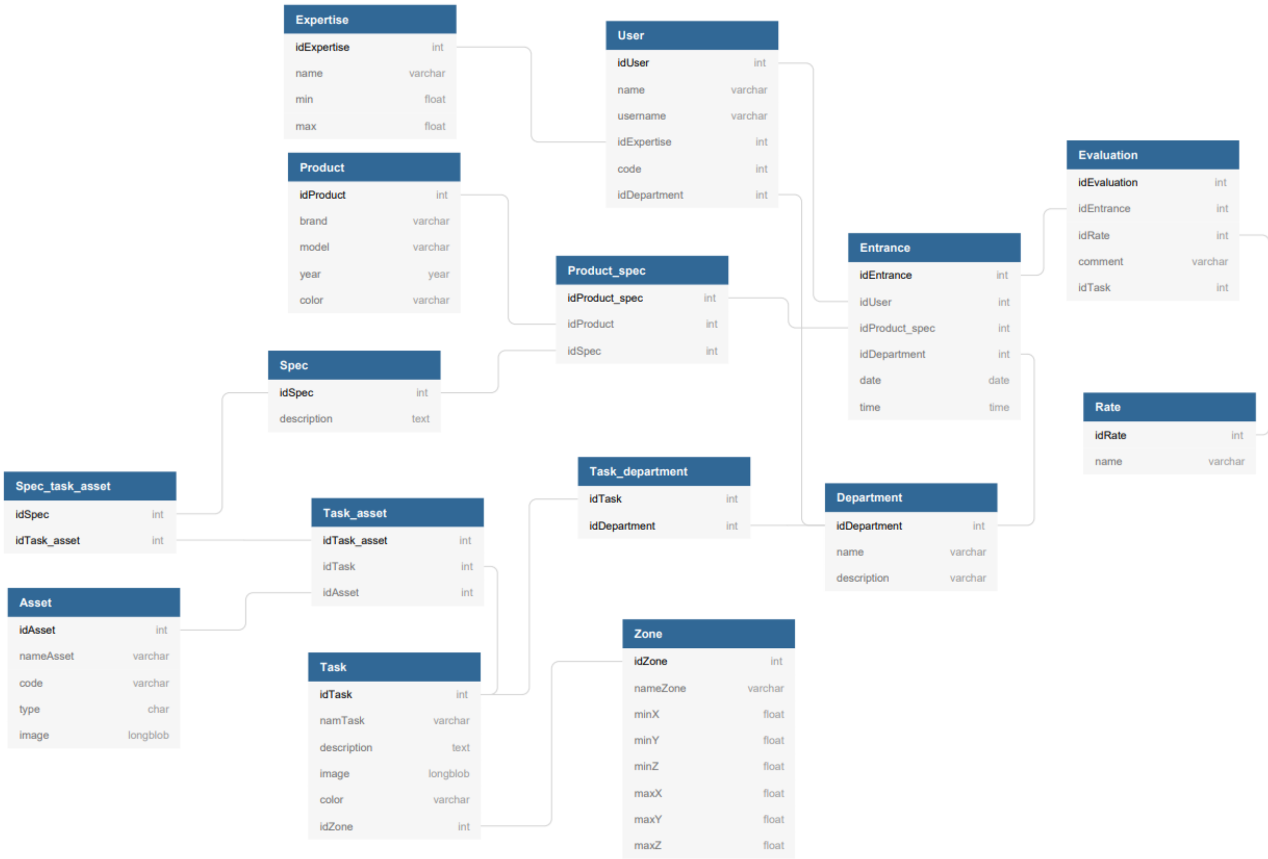


Figure 4.2: Database structure

The above Figure 4.2 is showing all the tables that compose the developed database and their relations. Bellow there is a more detailed explanation about each table's importance in the database as well as what it represents in the system.

- **Expertise** (**idExpertise**, name, min, max): Contains all the available levels of expertise as well as the minimum and maximum grade allowed for each level.

- **Department** (**idDepartment**, name, description): Represents one phase of the assembly line.

- **User** (**idUser**, *idExpertise*, name, username, code, idDepartment): Saves every user that may use the system, containing every needed information about them. The "code" field is used an easier way for the user to Login to the system.

- **Product** (**idProduct**, brand, model, year, color): Contains every product that cam be assembled in the production line as well as some interesting information about it.

- **Spec** (**idSpec**, description): Each Spec represents a group of tasks that has to be executed for a certain product.

- **Product_spec** (**idProduct_spec**, *idProduct*, *idSpec*): Makes the match between the product and the specs that it contains. Each product has one (i.e, a car with no extras) or more Specs (i.e, a car with every extra available),
- **Zone** (**idZone**, nameZone, minX, minY, minZ, maxX, maxY, maxZ): Represents every physical zone on the working area and it's minimum and maximum coordenates in a 3D representation.
- **Task** (**idTask**, *idZone*, nameTask, description, color): Contains every task that could be preformed at any stage of the production line.
- **Asset** (**idAsset**, nameAsset, code, type, image): Represents every object or tool that is required to successfully preform a task.
- **Task_asset** (**idTask_asset**, *idTask*, *idAsset*): Creates the match between a tasks and the assets. Each Task must have one or more Assets.
- **Spec_task_asset** (***idSpec*, *idTask_asset***, quantity): Links each Spec to the Task_asset that it contains. Each Spec must have one or more Task_asset, depending on how much Tasks a Spec requires.
- **Task_department** (***idTask*, *idDepartment***): Connects the Department with the tasks, listing which Tasks are preformed on each Department.
- **Entrance** (**idEntrance**, *idUser*, *idProduct_spec*, *idDepartment*, date, time): Register every entrance of a Product on a Department so that the system knows who assembled a certain product in order to be able to evaluate the success rate of each worker.
- **Rate** (**idRate**, name): Contains all the rate levels available for tasks evaluation.
- **Evaluation** (**idEvaluation**, *idEntrance*, *idRate*, *idTask*, comment): Register the evaluation rate given to each Task preformed on every entrance.

4.6 Algorithms

Each worker has an expertise level associated according to its success on executing the tasks that are demanded by each product on the production line. For this project, the task's evaluation is made using a vision system that is able to detect a set of colored objects on a defined area as it is explained on Chapter 5 when analyzing the system's architecture and on the subsection 4.6.2 bellow which explains how from an object position on space we were able to rate a task.

4.6.1 Experience Level evaluation

As explained before, on Chapter 1, the expertise level is calculated based on the worker's task completion success rate on a certain period of time or based on the number of performed tasks which can be gathered thanks to the two database tables created (Entrance and Evaluation) that keep track of when, who and what was assembled at all times.

For this case, it was defined that it would be used the information regarding the last day of work, and so, at a previously selected day of the month the following ordered steps are executed to get a new expertise level for the worker:

1. Get every task performed by the logged worker on the last defined period of time;
2. Group all the tasks by task;
3. Split all the grouped tasks into different arrays;
4. Calculate the average rate of each group of tasks;
5. Calculate how frequently each task appeared on the total executed tasks;
6. Apply the frequency to the average rate by multiplying them;
7. Sum every group's final average rate;
8. Get the Expertise level, from the **Expertise** table, in which the final result fits;
9. Update the worker's Expertise level.

At Step 1, the system starts by gathering every task that the worker was involved in for the past defined period of time which is accomplished by looking to the **Entrance** table and getting every entrance in which the worker was involved. After that, it looks for every task executed on those entrances by crossing them with the **Evaluation** table and with that it gets all the information regarding those tasks including the rate and how many times it appeared and in the same SQL query it groups them by task and calculates the average rate of each group as described on Step 2, 3 and 4.

For Step 5 the system calculates the total number of tasks by adding up the number of instances each task type appeared on the production line and then divides the number of appearances of each task type by the total number of tasks executed and this way we get a number between 0 and 1 representing the percentage of appearances of each task type on that period of time.

At this point the system already knows how many times a task was executed by a worker and what was the average evaluation for that task. The next step is described on Step 6 where the average rate and the frequency multiply and this happens so that we are able to apply different weight to each task success rate avoiding that one bad or good rate evaluation ruins the overall expertise level of the worker.

The final step to take, in an evaluation point of view, is number 7, where the system aggregates all the final results for each task so that we have a final evaluation grade to assign to the worker. After that we need to update the user expertise level by looking for the level that is appropriate for the final result assigned to the worker, and we do so by searching for the level that has a minimum value lower and a maximum value equal or grater then the assigned one.

4.6.2 Tasks evaluation

Although it is considered as a Disposable Variable on the Problem Definition in Section 1.5, to make it as similar to the factory environment as possible and take a step further by instead of doing a manual evaluation of the worker's tasks, making it automatic. We have decided to use an artificial **Vision System** developed by another team working on this same project to analyze the working area by recognizing objects in a Three-dimensional (3D) scenario and this way we are able to receive the objects position on the working area for further comparison with what is expected according to the tasks demanded for the assembled product in line.

4.6.2.1 Task

Before analyze how the task's evaluation was achieved, it is important to know what was defined as a task for the tested project.

The task is the actual action that the worker has to do and in order to test the developed system, there were created a set of **Tasks** and **Assets**. Since this system is using a Robotic arm and a Vision Camera (Section 5.1), there are a two restrictions imposed by these two systems:

- The Robotic arm is only able to carry up to 4kg of weight.
- The Vision camera is only capable of detecting Blue, Green or Yellow objects regardless their shape.

From these two restriction we have decided to 3D print a variety of blocks as light weighted as possible and then colored them using tape.

As a way to go along with these limitations and to fully test the system, a task is: **Place a number of colored objects in one of the defined Zones.**

Examples of this task description can be found on Figure 4.4 on Section 4.7.

4.6.2.2 Algorithm

At this point it is only important to understand that the system must receive a set of data that must be analyzed and inspected in order to be able to rate the tasks that were executed by the worker.

The data that the system is expecting to receive is a list of objects composed by the three three-dimensional coordinates (x, y, z) and a property named color that could take the values blue, green or yellow as shown in the example 4.1.

$$[\{x:0.2, y:0, z:1.2, color:'green'\}, \{x:1.1, y:0, z:0.1, color:'yellow'\}] \tag{4.1}$$

Later on the document, on Chapter 5, it is described that as a way to validate the proposal solution, the working area is divided into four areas where the user must place the requested number of colored cylinder. In order to be able to associate each received object's coordinates to one of the four delimited areas, the coordinates that narrow each area defined in withing the working area must be in concordance with the ones that the system has registered in table "Zone" of the created database shown in Figure 4.2.

The same way as the *Experience Levels evaluation algorithm* in Section 4.6.1 goes through a few steps to transform the task's rate into an experience level, this tasks evaluation algorithm also follows a structured sequence of events that will help us to rate each task by comparing what was expected to be found inside the working area with what we have actually received on the list objects (4.1) gathered from the Vision System.

1. Get each Zone's limitation coordinates from the database;
2. Check in which Zone each object on list is and adds an attribute "zone" with the proper letter;
3. Loops through the quantity of expected objects from each task;
4. Compare each expected object location and color with every one found;
   (a) If a match is found;
      - Add one to the count;
      - Deletes the object from the list of objects.
5. Rate the task based on the found count and the expected quantity;
6. Saves the evaluation to the database.

This process starts by getting the Zone's minimum and maximum coordinates for each axis (x, y, z) from the database and right away classify each object received from the Vision System with the letter of the zone it is located at which will be used later on when comparing with the expected ones.

Knowing that each expected task as a certain amount of Assets that must be placed on an indicated Zone, representing the number of times that task as to be executed, it is necessary to verify how many of those expected tasks where in fact performed correctly. For so, the algorithm compares each expected object's location and color with the ones captured by the Vision System, as many times as the task was supposed to be fulfilled.

If at any moment a match is found, i.e an object's position and color, on the working area, correspond to an expected task, a counter variable is increased by one and the found object is removed from the list of objects gathered by the Vision System so that it is never used again as a comparison to another task.

Finally, the task is rated in a predefined scale when comparing the expected quantity of objects to the ones found.

4.7 User Interaction

The interface is the main aspect of the presented dissertation as it is the bridge between the Human and the remaining of the system, it is the machine with which the Human communicates and interacts.

As a way to identify the worker that is interacting with the interface, a simple login page (Figure 4.3) was created where the worker has to insert a four digit login code that represents him in the system and from this point forward the system has all the required information to present the correct interface and the tasks that the worker is capable of executing.

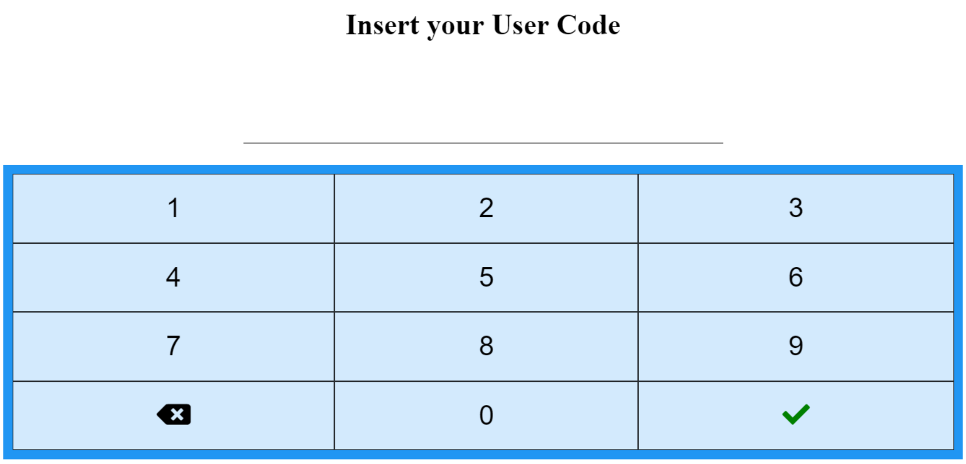


Figure 4.3: Login Page

After the login is successfully terminated, the system presents the user with the main screen where everything is introduced.

Based on what was found on Chapter 2 the developed interface is mainly composed of "cards" that contain the information regarding each task that the worker must perform as Figure 4.4 demonstrates.

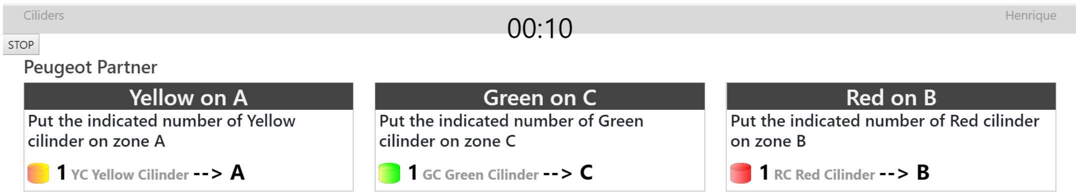


Figure 4.4: Tasks Page

The interface contains a top bar containing complementary information such as the Station in which the worker is working on and the stopwatch indicating how much time the worker has to finish the tasks for that product. Right below this top bar there is a section with a more relevant information about the product in construction showing the product's name and extra features if the product demands so, also in this section is placed the STOP and RESUME button explained on the next Chapter on Section 5.5 .

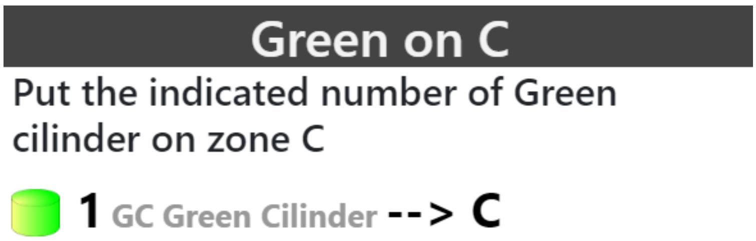


Figure 4.5: Card with information about the task for a Novice worker

These "cards" (Figure 4.5) are essentially a rectangle where all the necessary information about the task is presented, containing the task's title on the top part of the card, the task's description right bellow the title where it is explained in a more extensive version how the task must be performed and after this description, a list with the necessary Assets is presented, indicating the quantity, code, name, image and the Zone's letter in which the worker must place the Asset if applied.

4.7.1 Grab User's Attention

Chapter 2 explains how one can grab the user's attention through an interface and using the acquired knowledge, Figures 4.6 and 4.7 where designed and developed in order to grab user's attention making sure he is aware of what is happening on the system.

Figure 4.6: Card with information about a task that is only present when an extra package is requested

The above figure represents a task that has to be executed and that is why it has the same structure as regular task, the only difference between the Figure 4.6 and Figure 4.5 is how frequently they appear on the screen. For this project it was defined a Default set of tasks that must be performed for every product that may show up for assembly and those tasks are presented as Figure 4.5 demonstrates whereas whenever a product has a specific set of tasks that only belong to that product they are presented as Figure 4.6 by changing the border color to red that represents attention and caution and the title's background adding a colored patterned as a way to enhance their appearance and grab the worker's attention to that new and different task that is crucial for that products overall correct assembly.

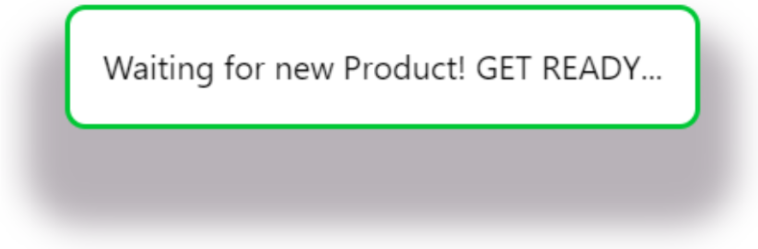


Figure 4.7: Message shown when waiting for the new product in line

While the Vision system analyze the working area and the Robotic Arm is reaching out for the new box of assets, the worker is presented with the Message that Figure 4.7 is showing. This happens so that the user is aware that the product is complete and a new one is being prepared to get assembled and by clearing the interface, we are able to attract the user to the new information. This message is presented with a fat rounded green border passing out a feeling of success and achievement, note that for this message it is used the Depth and Size illusions to get the user to absorb the most important information which in case would be the "GET READY" part of the message, both illusions are explained earlier on this document.

4.7.2 Adaptive

As it has been referred throughout the document, one of the goals of this proposal is to develop an interface that is able to adapt to the user's expertise level and that is accomplished by assigning and interface to each level of expertise available which for the test's environment it was defined as being three levels.

Although these three interfaces vary from one another, their changes are minimal keeping the fundamental structure intact avoiding a contrary effect from the one it is aimed for. This minimal changes rely on the format that the information is shown to the user regarding the assets as it is the only type of information that is flexible to changes.

Table 4.1: Format of information available for each level of expertise

	Image	**Name**	**Code**
Novice	Yes	Yes	Yes
Intermediate	No	Yes	Yes
Expert	No	No	Yes

Table 4.1 gives us information of what is the format of the task's information that is presented for each expertise level. As it is visible, the **Novice** user is the one with less experience which means that it is the one who needs more support and so for each asset that is necessary for a task the card holds an image, the name and a code that represent the asset. For a **Intermediate** user, the asset is described with the name and the code (Figure 4.8), finally for an **Expert** user, the asset is only represented by its code (Figure 4.9).

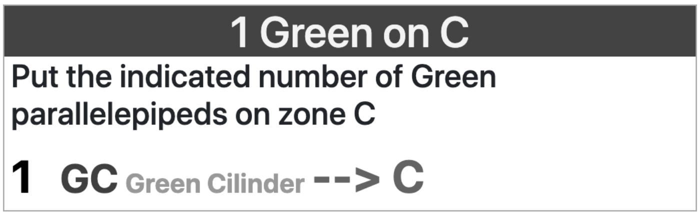


Figure 4.8: Card with information about the task for an Intermediate worker

The figure above shows how the cars look like when a worker is considered Intermediate. The major change from this card to the one presented to a Novice is the absence of the image which straight away feels like a much less attractive interface, yet, at this point the worker should be able to identify the assets by its code and name alone. Note that for this experience level, the asset's code "GC", is larger and darker than the previous one, making it easier to identify. The name is still present, to work as a second level of support, and it is noticeably smaller as a way to train the worker to get used to identify the asset only by the code, which is expected of Expert workers. Finally, for this level, the zone's color was brightened slightly as a way to enhance the remaining information.

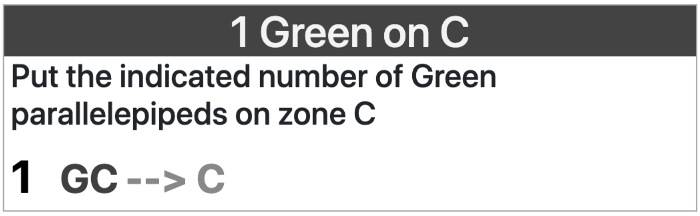


Figure 4.9: Card with information about the task for an Expert worker

For Expert worker's the amount of information disposed is as minimal as possible, Figure 4.9. The only two changes from the intermediate level are the absence of the asset's name as it was expected and the zone is smaller and brighter in order to increase contrast and could possibly be removed if there was a higher expertise level.

4.7.3 Cognitively light

In order to make all of this system's interfaces as light as possible for the worker, the list of suggestions given at Section 2.4 on the previous chapter was taken into consideration and used as a guide to design them. The interface contains a clear core area that contains the most relevant information and instructions which are simple and concise, as introduced before, although there is a change on the interface regarding the level of expertise of a user, the format is consistent throughout the different interfaces either if the expertise level changes or if the tasks are unique for a product.

As to color schema, the taken approach was to keep the most relevant information with a darker color and the less important make it clear while having a white background.

Chapter 5

Test environment

Contents

5.1 Introduction

This dissertation is inserted in the scope of a larger project being labored by the whole investigation team and so, as a way to simulate an assembly line section as can be seen in an industrial environment, a small CPS was developed using the previously described interface, a robotic arm coordinated by another element of the team and an artificial Vision System handled by a third person. This chapter describes how we were able to get as close as possible to a real world environment justifying the appearance of every intervener on the system. It is also here that it is explained how the tests were conducted and what was their goal in order to understand how easy it is to interpreter the interface and how well were the algorithms implemented.

5.2 Nomenclature

In this section we present a set of terms that will be used in the remaining of the Chapter that are considered important to better understand the constructed scenario to test the developed solution and simplify its explanation. Some of the definition presented on the table 5.1 were already presented on the previous chapter when introducing the database and all its tables and fields.

Table 5.1: Nomenclature

Station	Where a specific set of Tasks is performed
Task	What the worker has to do
Asset	Physical instrument/object needed to execute the task
Product	The expected result by the end of the assembly line
Default	A set of tasks present in every product
Extra	A set of tasks that need to be executed for a specific product
Time Period	The time given to the worker to execute the set of tasks needed for the presented product
Stop	Button pushed by any worker on emergency that stops the entire production line
Expertise	Level of expertise of each worker
Evaluation	Where the final product is examined to guarantee it's quality by rating every task executed during production
Rate	Mark given to every task in a specified scale

5.3 Architecture

The architecture created for the solution is divided into three main components, easily identified on Figure 5.1. Each component as a smart component (Digital Twin) that represents it in the digital world. There is also a forth smart component that is responsible for the connections between all the other three components and controls the action flow of the system, nominated **Process Smart Component**.

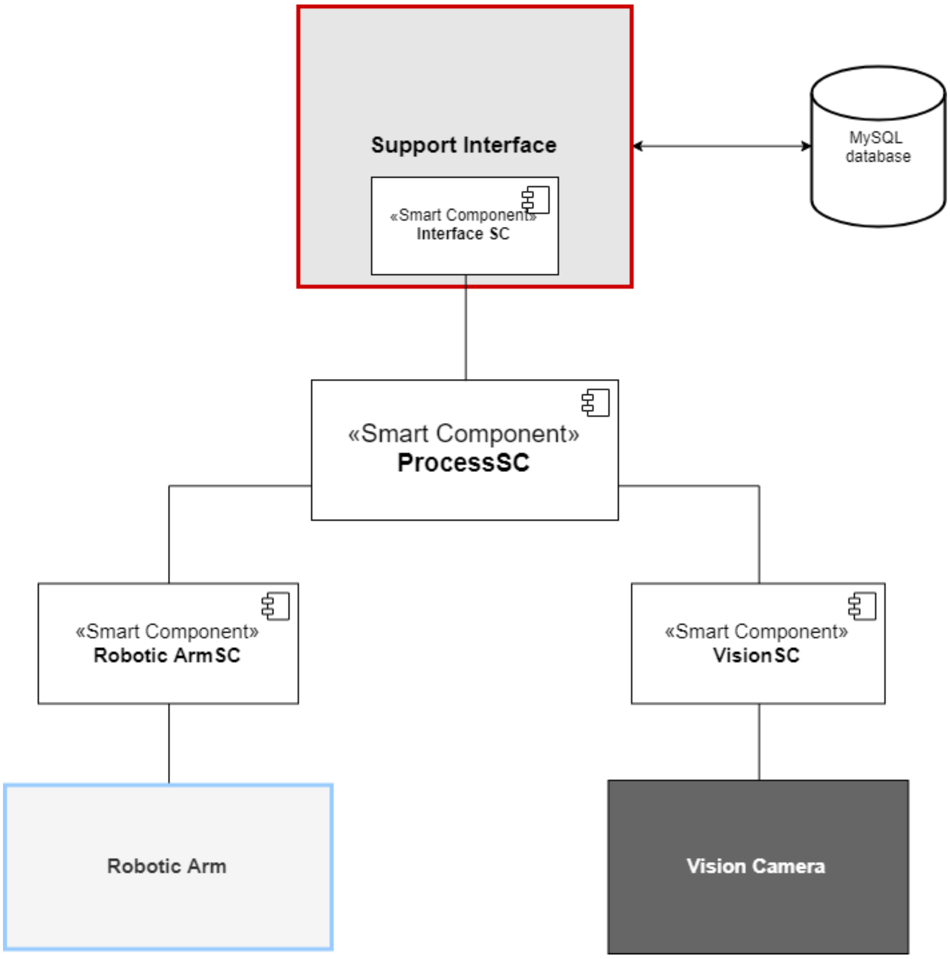


Figure 5.1: System's Architecture

The first component is the **Support interface** developed with which the worker interacts where the tasks are displayed as the product progresses in the assembly line. The communication with the smart component happens every time a new product enters the station where the smart component sends an Id that represents the product, as well as the Time Period in milliseconds and the support interface communicates with a database where every task needed to be executed is provided and then displayed to the worker.

The second component is the **Robotic Arm** whose job is to deliver to the worker a box with every asset necessary to complete the tasks. For this component, the smart component is given the box position and the final position and the arm picks up the box and drops it at the final position where the worker starts to preform the tasks.

The final component present in the system is the **Vision Camera** that is responsible for verifying the tasks executed by the worker and rate them in a specific scale with which the algorithm will calculate the expertise level of the worker and consequently adjust the interface to the user.

5.4 Restrictions/Frailties of the CPS

As everything in the world, this CPS also has a few restrictions and frailties from each component on the system that limit the tests and consequently delay the evolution process demanding a higher number of tests and participants since the test's complexity is also limited.

Interface:

- Requires user input from external mouse controller;

Robotic Arm:

- Maximum payload weight of 5 kilograms;
- Maximum reach 0,85 meters;

Artificial Vision system:

- Only able to detect Yellow, Green and Blue objects;
- Not able to distinguish objects by their shape;

The described restriction are related to the current state of the CPS which could be improved with extra implementations.

5.5 Test Case

This section explains how exactly the tests work by describing all the elements present on the test, the flow of events necessary to correctly perform the test, how the CPS is behaving while the user is testing the system, and finally some variables that were defined beforehand.

5.5.1 Scenario

1. Worker joins the Working Area
2. Worker enters user code
3. Robotic Arm moves assets box to near the worker
4. Interface shows tasks
5. Work time starts
6. Worker reads tasks
7. Worker executes tasks
8. Work time ends

9. Camera analyze the area
10. Repeats processes 3 to 8

5.5.2 Connections

Now that the actions flow for a proper test performance was presented we may go a step further and dig a little deeper on understanding what is actually happening "behind the scenes" concerning the communications and logic associated with the previously described actions flow.

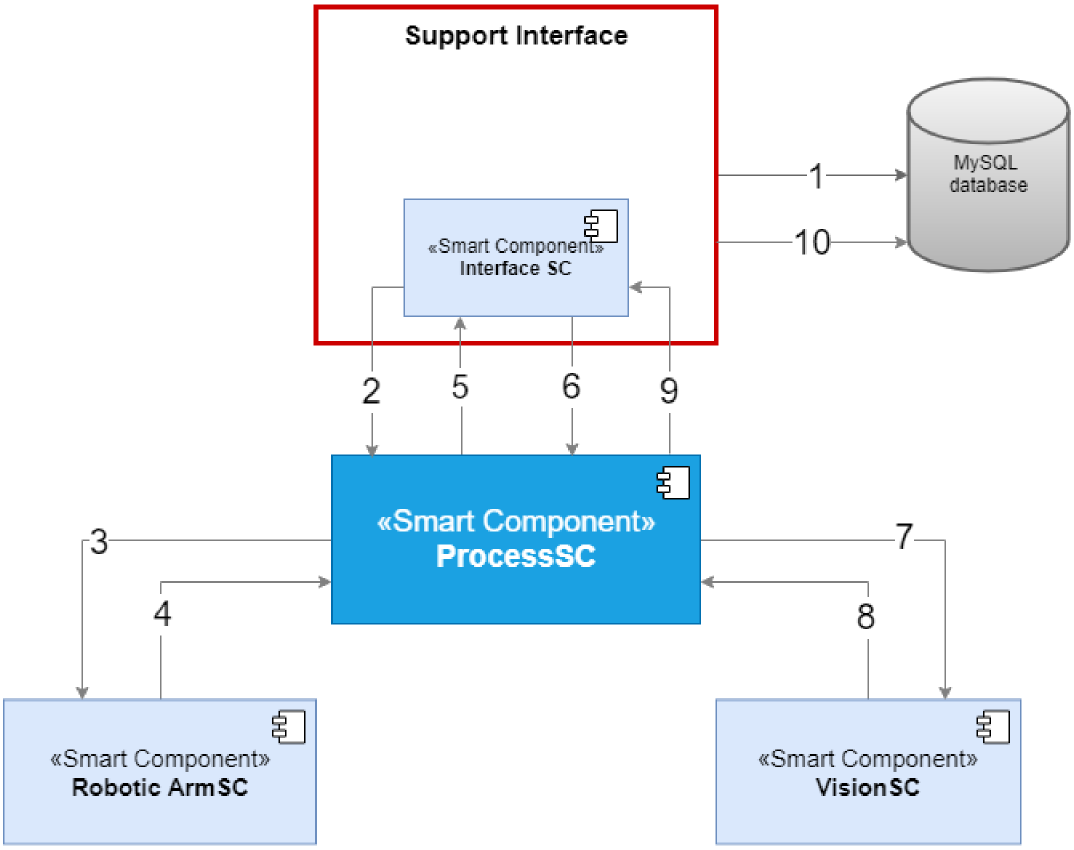


Figure 5.2: Flow connections between smart components

The above Figure 5.2 is composed of all the components of the system that are evolved with the communications that happen during the test execution, each arrow represents one of those communications and it's numerations indicate the data flow throughout the test case.

1. User Login;
2. **Interface** tells the **Process Component** that the user is ready and the system may start;
3. The **Process Component** tells the **Robotic Arm** to start the movement;
4. The **Robotic Arm** informs that the movement is complete;

5. The **Process Component** tells the interface which product is being assembled and how much time the worker has and so show the tasks to the worker;
6. The **Interface** indicates that the time has terminated;
7. The **Process Component** tells the **Vision system** to start analysing;
8. The **Vision system** answers with the objects position on the working area;
9. The **Process Component** passes it out to the **Interface**;
10. The **Interface** saves the evaluation rate of the tasks after comparing the objects real positions with the expected ones.

Besides these, there are two other connections that can happen at any time during the system's execution, and they are activated by the worker when the Emergency button is pressed. This button is implemented on the Interface to simulate the Emergency Stop Handle that is available throughout the entire assembly line as a last minute resource if the worker feels that he will not be able to complete all the tasks in time and the whole system must freeze. For so there are two complementary connections between the **Interface** and the **Process Component**:

1. The **Interface** indicates that the emergency button was clicked to STOP;
2. The **Interface** indicates that the emergency button was clicked to RESUME

5.5.3 Defined Variables

Similar to Section 4.4 on the previous Chapter, where there were defined three variables considered influential to the final solution and if their values where to change, the solution's outputs would be different, also here are some other influential variables that where defined for test purposes due to time, resources and people restrictions.

- Time Period analyzed: 1 day;
- Type of information: Image, Code, Name;

5.6 Points of Evaluation

The most important point of evaluation for the overall well function of the system is the rate given to each performed task by the user which is gathered by the Vision System, besides the rate, it was also registered the time that it took to complete a set of tasks.

To test the developed system, there were performed tree different tests, a **Logical Test** that is used as a way to test how easy and intuitive it is to use and learn the interface, a **Control Group** where the goal was to understand how much the interface helps to execute the tasks and a **Algorithm Test** that had the purpose of testing how good and accurate the developed algorithms perform.

5.6.1 Logical Test

The Logical Test aims to understand how easy and intuitive it is to use and learn a software by comparing the results of its usage and success when performing the tasks that the applications demands with the results acquired with a previously completed logical test. With this test it is possible to evaluate a variety of aspects and variables, for this project it is used to compare the success rate and time needed to complete both the logic test and the application's tasks. With this comparison it is possible to conclude that, if for one individual user the results are similar, the application is good and easy to use or on the other hand what should be changed on the application to make it easier and more intuitive.

As a credible logical Reasoning test it was used the 123test website, based in the Netherlands, which provides free practice tests. With over 900,000 tests completed monthly, for the last ten years, 123test as created and published tests online that answer questions about the IQ, personality and career assessment [1].

5.6.2 Control Group

The Control Group is composed of participants that are not receiving the experimental treatment [6] that for this case was the use of the interface when performing a set of tasks. This group of participants play an extremely important role in the research process, as they serve as a benchmark to understand what is the impact of the independent variable, comparing their results with the ones obtained by the users who used the interface to execute the same set of tasks [6].

To get this type of environment where the interface is not used to execute the same tasks, using the same working area, and the same assets, the tasks were shown to the participant in an Excel document as a list and there were registered the average time on task and the number of tasks successfully executed. With these values it was possible to compare them to the ones gathered with the participants that used the interface and conclude how much the interface helped the user.

Chapter 6

Results

Contents

6.1 Introduction

In this chapter are presented the dissertation's results in form of graphs that assesses the tests performed to a group of users. There is also explained what would be the expected results if there were built more tests and possible sets of tasks to be tested by changing their complexity or quantity per product. Note that these are merely expectations and not conclusions as those tests were not executed. As a way to keep a coherent format throughout the chapter, the results from each test are represented with a different color as a way to easily identify from which test the results on the graph belong to, therefore, the Logical Test is associated with the color Green, the CPS with Interface with the color Blue and the CPS without the Interface with Red.

6.2 Users

The set of users presented to these tests are mainly Engineering students aged between 20 and 25 as this system was part of a larger product and due to the lack of mobility from the Robotic arm and the Artificial System, most of the tests where performed at FEUP facilities which reduced the variety of possible users. However, some of the tests that do not demand the use of the developed support interface were done on a separate location making it easier to get to different type of users. The overall number of participants was 13 with around 62% of were male and 38% female.

6.3 Tasks

Before getting into detail on the tasks to be performed, and in order to have a more visual perspective about these tasks, let us present with the assets used to execute the tasks (Figure 6.1) and the working area divided in four smaller areas (Figure 6.2).

Figure 6.1: 3D printed box with 3D printed colored blocks

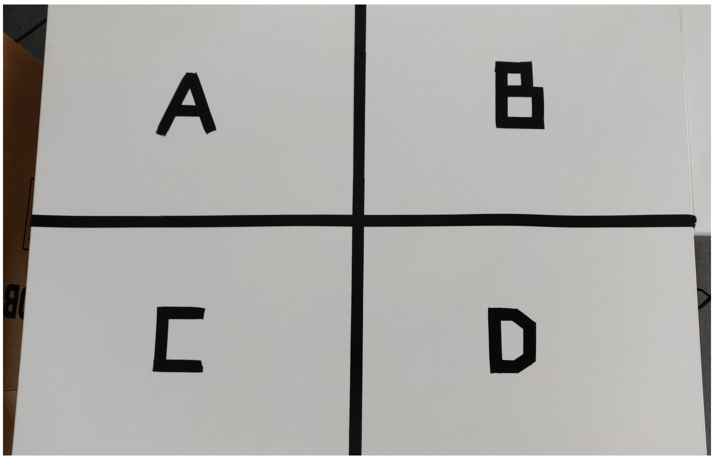


Figure 6.2: Working area divided into 4 areas

As it was explained before, for each product on the assembly line there are always a set of tasks that are considered as default from product specifications and there can be a few extra tasks belonging to an extra specification of a particular product, and so for these tests there were created five different products summing up five possible combinations with the Default Spec and the other four specifications, remembering that every product should have the Default Spec and could or not have one Extra Spec.

1. **Default:**
    - Put 1 Yellow parallelepiped on Zone A
    - Put 1 Blue parallelepiped on Zone B
    - Put 1 Green parallelepiped on Zone C
2. **Yellow:**
    - Put 2 Yellow parallelepipers on Zone C
    - Put 2 Yellow parallelepipers on Zone D
3. **Green:**
    - Put 1 Green parallelepipers on Zone A
    - Put 2 Green parallelepipers on Zone D
4. **Blue:**
    - Put 1 Blue parallelepipers on Zone A
    - Put 1 Blue parallelepipers on Zone C
    - Put 1 Blue parallelepipers on Zone D
5. **Full D:**
    - Put 2 Yellow parallelepipers on Zone D
    - Put 2 Green parallelepipers on Zone D
    - Put 2 Blue parallelepipers on Zone D

The Extra specs were defined taking into consideration that a task present in the Default Spec could not be seen on any Extra Spec as it can be verified with the list above. For each product, a time limit in which each user should complete the tasks was defined. The table 6.1 bellow shows which Specs belong to each product, how much time was defined to each Spec and the total time attributed to the product.

Table 6.1: Products Specs and total time available to perform each product's tasks

	Specs					
	Default	Yellow	Green	Blue	Full D	
Time (s)	15s	+5s	+15s	+10s	+20s	Total time (s)
Product 1	X					15s
Product 2	X	X				20s
Product 3	X		X			30s
Product 4	X			X		25s
Product 5	X				X	35s

As explained before, every product must have at least the Default Spec and an optional Extra Spec. For these tests there were only created products with no more than one extra Spec due to the number of assets that were printed. Additionally, the developed test flow demanded that the box of assets had to be prepared before the next product would appear, which did not allow for a product with more than one extra Spec. The table 6.1 demonstrates that participants always had at least 15 seconds to complete every task and an extra time, defined according to the level of difficulty of the tasks, would be added depending on the extra Spec belonging to the product in assembly, giving a total time to complete the tasks marked on the last column of the table.

6.4 Logical Test Results

The logical test was composed of 10 questions where the users should select the forth image of a sequence being given the first three, and this way, guess what the next one should be. The Figure 6.3 shows a graph of the results gathered from the 13 participants. The main objective of this test was to gather information regarding the user's ability to understand and identify what to do when given a pattern, and compare afterwards with the results when utilizing the CPS with the support interface.

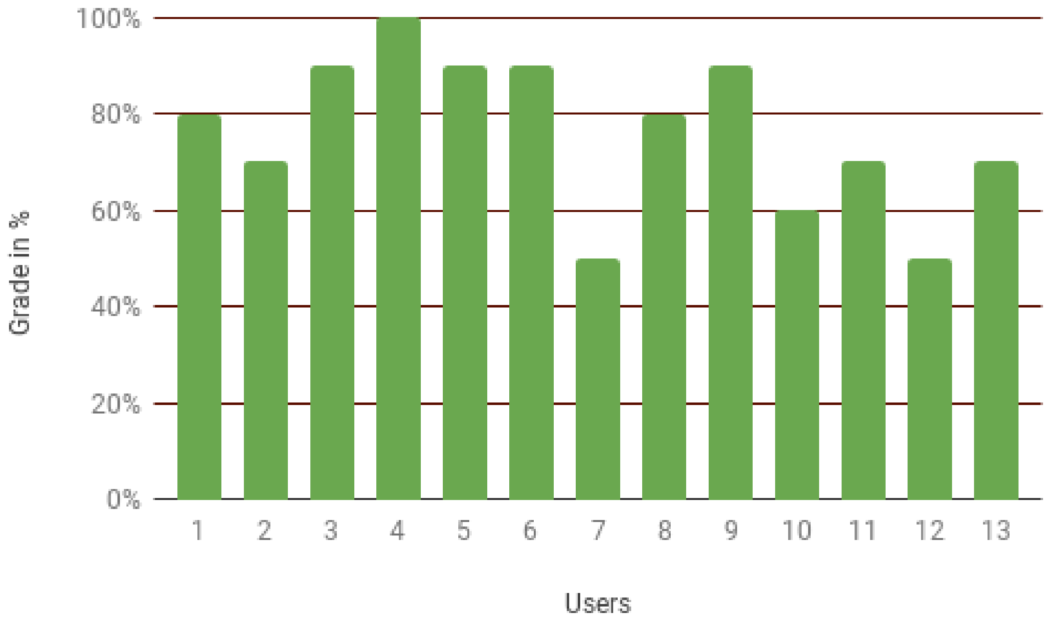


Figure 6.3: User's grade in percentage on the logical test

By looking at the graph it is possible to identify that every participant got 5 or more correct answers out of the possible 10 and a little over 50% of the users had a final grade of 80% or higher, meaning that they got at least 8 correct answers but only one participant answered all of the questions correctly. With a little more analysis it possible to indicate that the average grade of the group was of 76%, with a standard deviation of 15%, which is understandable due to the difficulty of some of the questions and the fact that there were only 10 questions, increasing the pressure of each question in the final result. It is also important to mention that for these tests there was no time limit to finish the test and the average time spent by the group on each question was around 40 seconds with the fastest average time of one participant being 22 seconds and the slowest taking 67 seconds.

6.5 CPS with the Interface

The results shown on this test were gathered from a group of 10 participants that used the CPS as it was described earlier with all of the components involved. Figure 6.4 shows the results in a success rate presented in percentage per user when executing the presented tasks.

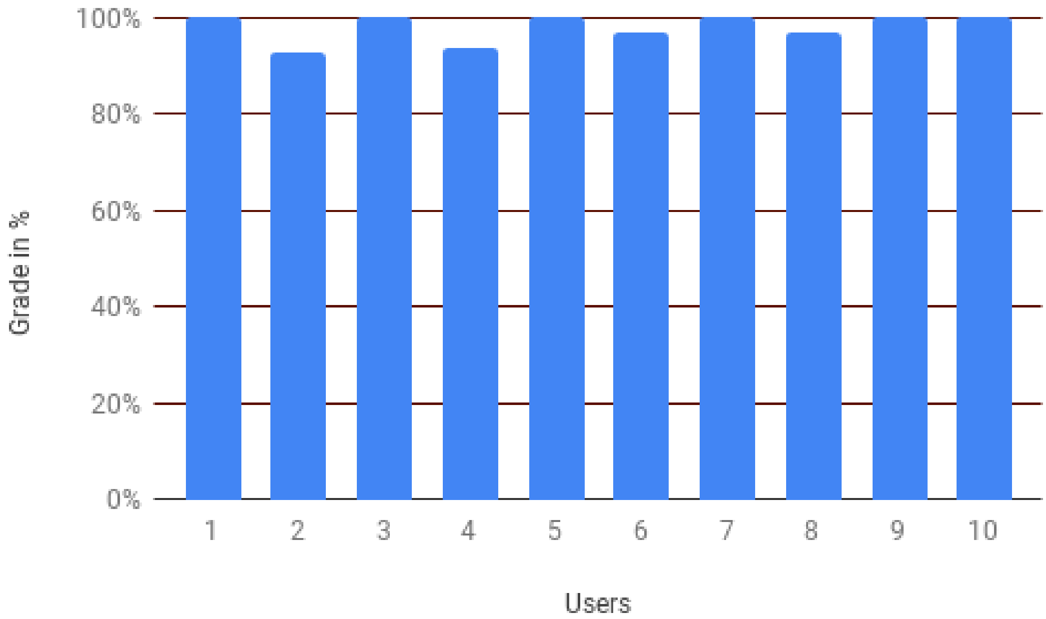


Figure 6.4: User's grade in percentage when testing the CPS with the Support Interface

Analyzing the above graph, it is visible that every participant got a mark higher than 90% and 6 out of the 10 participants executed every task correctly, giving us an average of 98%, with a standard deviation of 2,7%. Looking at these results alone and without comparing them to anything else, there are a few prepositions that come to mind:

- The tasks were very easy;
- The interface is very helpful and easy to understand;
- The user had too much time to perform a group of tasks;
- There were very few tasks to execute on each product;
- The participants were very clever.

To confirm these prepositions it would be necessary to perform a few more tests that counter these prepositions, that is, create harder tasks, remove the interface, reduce the time to accomplish the tasks, add more tasks on the same period of time and perform these same test with a different group of participants, or perhaps having participants in a higher age range.

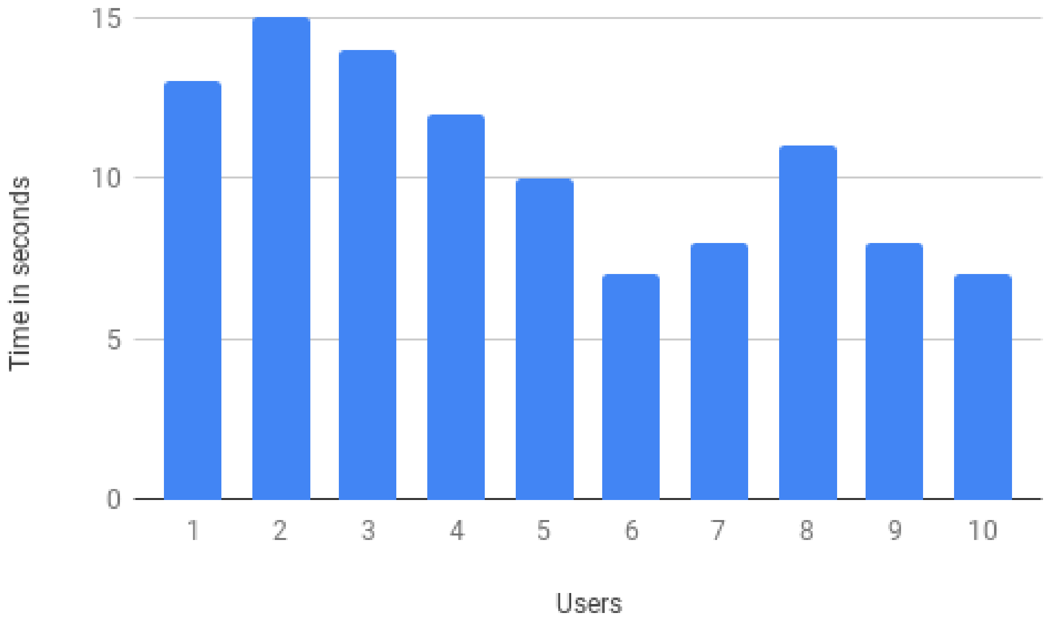


Figure 6.5: User's average time spent on each product's tasks with the Support interface

Figure 6.5 shows how much time in average a participant spent on each product and it is immediately visible that no participant took more than 15 seconds in average to perform the tasks. This evaluation leads us to believe that the time attributed to each product (Table 6.1) was probably too much as the minimum time possible for a product was of 15 seconds and, in average, participants used 10.5 seconds, which could be one of the reasons the success rate was so high, as pointed before.

6.6 CPS without the Interface

The results presented on this section belong to the tests done with six of the thirteen participants, and consisted by executing the exact same tasks as the group presented on the previous section, where the only difference was that these participants were shown the tasks on an Excel document as form of list, with only the description of the task, similar to the list of tasks exhibit on Section 6.3. This test was created to serve as a control group to the variable being tested, in this case, the Support interface, this way it is possible to compare the participants performances so that it is possible to validate the necessity of the support interface in this type of environments.

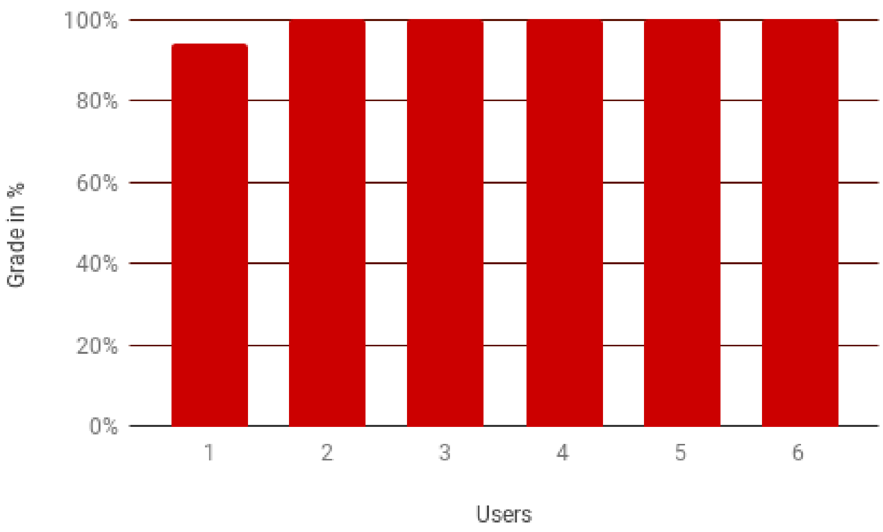


Figure 6.6: User's grade in percentage when testing the CPS without the Support Interface

Once again, on Figure 6.6, it is visible that the majority of the participant on this case had a grade of 100% and the average result of the group is 99%, with 2% of standard deviation, which could be justified by the same reasons pointed out previously. It is also important to mention that 3 of these 6 participants, performed this test after performing the one with the Support interface, which could influence these high results as well as the time spent on task, shown on Figure 6.7.

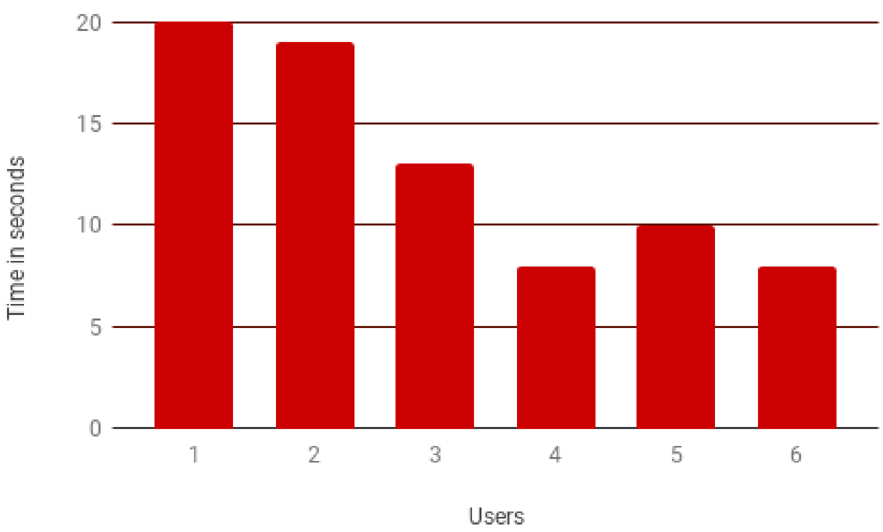


Figure 6.7: User's average time spent on each product's tasks without the Support Interface

By looking at the chart above and comparing the three first participants with the three on the right-hand side, we identify the three participants described on the previous paragraph as their performance's time is fairly lower than the three users that only used the CPS without the Support Interface.

6.7 Comparing Results

During this section there will be placed side by side the results of all tests and compare the results to draw some conclusions about the proposed and developed solution. This way, it will be possible to identify the strong aspects of the solution and which aspects need to be redefined and retested, starting by comparing the results from the Logical Test and the CPS with the Support interface, and after that compare the results from the CPS with the Support Interface and without the Support Interface.

6.7.1 Logical test vs Interface

The graph shown on Figure 6.8 exhibit the results gathered from the same participants in both Logical Test and CPS with the Interface so that it is easier to analyze and compare the participants performances on both tests.

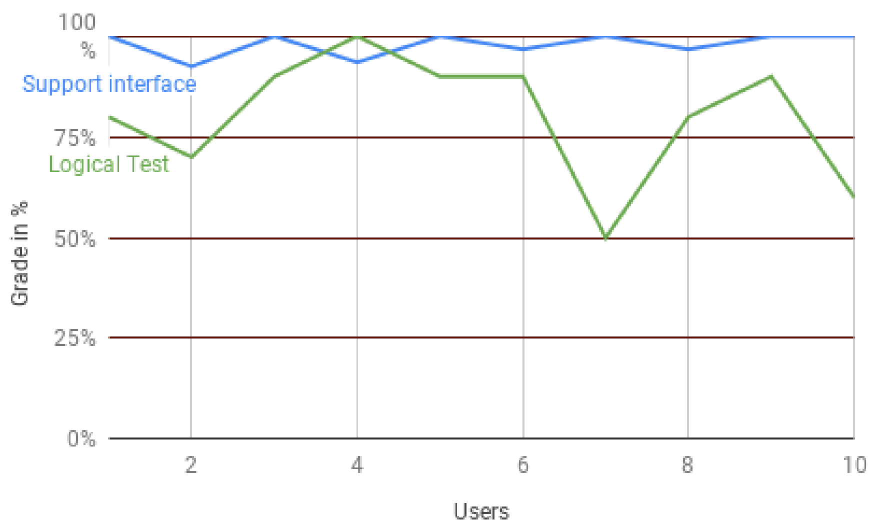


Figure 6.8: Graph with the results from the logical test and the CPS with Support Interface

This comparison is created as a way to understand what is the meaning of the successful results gathered from the participants that used the CPS with the interface. This way, it will be possible to answer one of the prepositions pointed in Section 6.5, saying that high grades mean that participants are very intelligent or if the interface is actually helping the user.

If the results from the Logical test were higher than the ones from the CPS, it would mean that the interface was neither intuitive nor helpful, whereas if the opposite situation was found or the results were the same or very similar, it would be possible to conclude that the high results were not only due to the level of intelligence of the participants but that the Interface is actually good and helpful.

Straight away we are able to conclude that the performances from the CPS with the Support interface, presented by the blue line on the graph, were better than the ones on the Logical Test, displayed by the green line, and the proposed Interface is good and led to a low rate of mistakes.

6.7.2 Interface vs No Interface

In these two test comparison is where we look simultaneously at the results from the Control Group, described on Section 5.6.2 in Chapter 5, which is the group of users that tested the CPS without the Interface and the Experimental Group that tested the independent variable, the Support Interface. In the next sections we will be looking at the **Grade** and **Time** separately as their results comparison give us different perspectives of the Support Interface influence on the worker's performance.

6.7.2.1 Grades

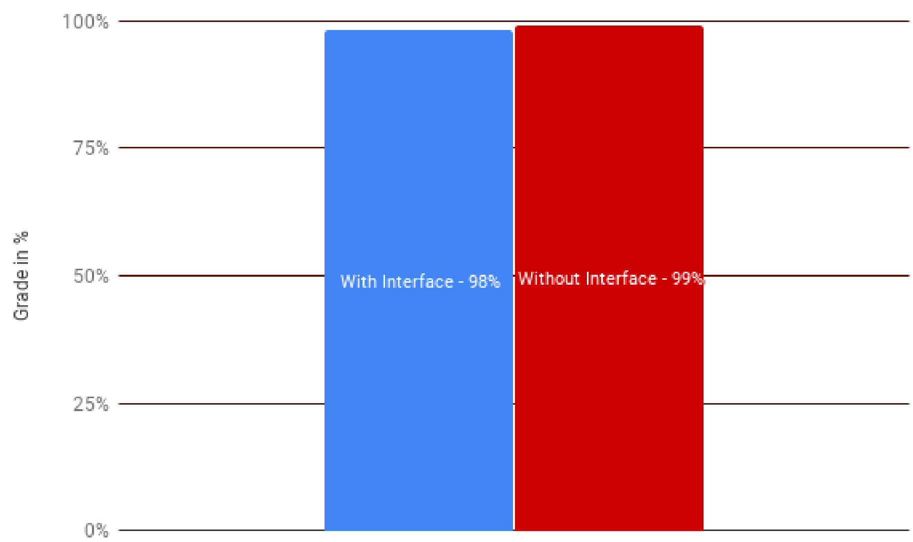


Figure 6.9: Chart with the average grade, in percentage, from the CPS with and without the Support Interface

If we were to look at Figure 6.9 as a choice to make, everyone would choose the option on the red column labeled "Without Interface" as it has a higher percentage than the blue on the left side, yet in this situation this discrepancy is close to zero as the number of tasks performed with the Interface is higher than the ones performed without the interface which increases the error probability. One conclusion that is possible to get from this graph and comparison is that the tasks were actually quite easy and the Interface could not be proof to be helpful on making them even easier to perform as they did not demand any complex combination of Assets or instructions.

6.7.2.2 Time

Now that the conclusions taken from the Grade perspective of the results when comparing these two tests have not given any significant insight regarding the Interface as a key factor to the workers performances, we will now analyze the results using the Time spent on product assembly, shown on Figure 6.10, as a comparison between them.

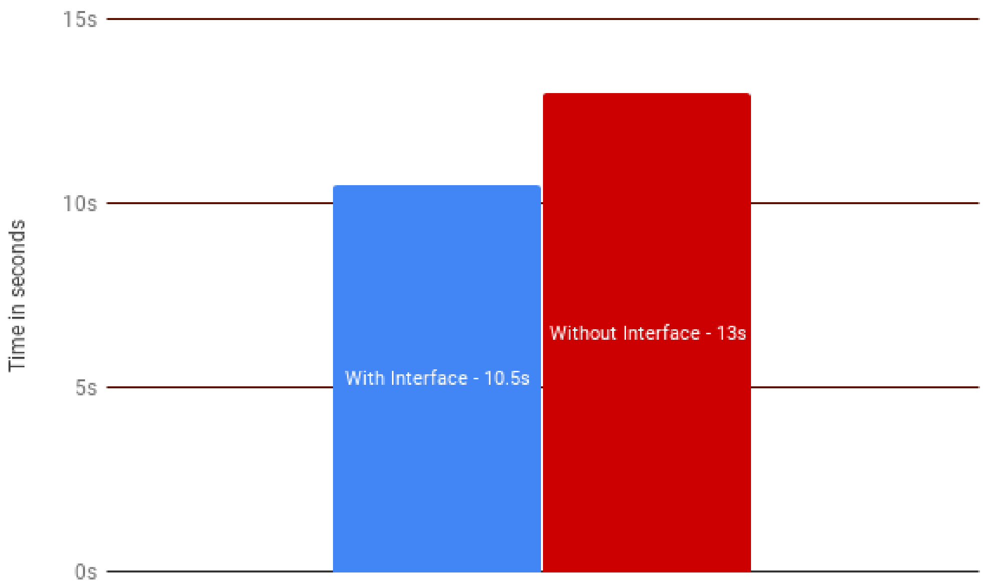


Figure 6.10: Chart with the average time, in seconds, spent on each product, by the participants when using the CPS with and without the Support Interface

This Figure shows that the average time spent on a product when using the Support Interface was of 10.5 seconds whereas without it, the participants took around 13 seconds to complete every task. Taking the exact same approach as on the Grade analysis and look at the graph as a choice, in this case, choosing the blue column is even easier due to the evident values difference. Oppositely to the Grade analysis, the discrepancy detected in this graph is now significant and must not be ignored as it leads to believe that the Support interface presence on the system had a positive impact on the participants performances.

Knowing that a tested product is composed by three to six tasks, and the average time spent on these tasks was of 13 seconds, this would mean that for each task the participants took between 2.16 and 4.3 seconds on each task. Hence, having a discrepancy of 2.5 seconds between these two tests could mean that, if the time allowed on each product had a small margin for error, without the Interface, there could not be enough time to complete the tasks.

With this, we conclude that the worker actually benefits from the Support Interface as it is easier to understand and keep track of which tasks the worker is performing.

Chapter 7

Conclusions and Future Work

Contents

The present dissertation has its roots on a challenge detected by the PSA Group of Mangualde regarding the cognitive load of the factory's workers, attending the graphical interfaces used, which was reducing their performance levels and affecting the factory's vehicles production. Hence, this dissertation aims to address how the information is presented to the assembly line workers as they assemble the vehicles, splitting this question into two problems. The first one worrying on how to present this information as cognitively light as possible, while nowadays it is shown on a sheet of paper with no proper reference on which assets to be aware to use or install on the product on production.

The second problem is related with information being shown the same way to every worker and the same sheet of paper passes by every section of the assembly line, not taking into consideration each worker's specific needs and disabilities nor expertise level, as it does not matter if it is the operator's first day on the job or tenth year on it.

As a deeper research took place, a possible solution started to be molded and the two problems started to merge and depend on each other. Designing and developing an adaptive interface according to the expertise level started to be influenced by the way information was supposed to be arranged and structured in order to reduce the cognitive load of the worker.

Finally, a solution was developed consisting on an interface that was capable to change the format of the information displayed depending on the expertise level of each worker, that is calculated based on their success rate over a period of time, and the color pallet and font size used aims to stand out the most relevant information using the color, size and depth contrast principals to grab the worker's attention highlighting the information that helps the user to quickly complete the tasks.

7.1 Contributions

As it was defined on Chapter 1, the main objective of this dissertation was to have a positive impact on the assembly line workers by:

- **Providing the worker with an interface that is able to adapt itself to the user's expertise and reduce cognitive load.**

As explained before, most of the work developed belongs to a problem stated by the PSA Group of Mangualde and so, all of the implementations and tests were created in a small scale through a prototype system as a way to try to validate a possible solution for the identified flaw on the industry regarding the cognitive load of the assembly workers. Despite it not being implemented in an industrial environment, the developed CPS and solution was important to validate how much a Support Interface like the one presented could contribute for the factory's workers as well as comprehend how systems like this can benefit from it.

The Support interface developed allowed us to understand that it is possible to have an interface that actually supports the worker by providing the amount of information that the worker need according to its capabilities and expertise by replacing the current system seen in the factory floor to the proposed one. It also leads us to believe that machines are able to adapt to humans making them the support actor instead of the main one.

These solutions also proved that a business can strongly benefit from this approach by only testing it in a small scale environment, where the average time on product assembly may increase at least two seconds on an easy and quick set of tasks.

In parallel to the dissertation it was also developed a scientific article with the same scope that was submitted and accepted to The Eighth International Conference on Intelligent Systems and Applications - INTELLI 2019 - that takes place in Rome, Italy [22].

7.2 Applicability

All of the proposed solution was developed having in mind the assembly line structure found in the PSA Production Unit, yet, other kind of systems would benefit from this Support interface as it can be applied to any environment that involves a sequential process that contains different sections each one with tasks to be accomplished.

For instance, in a kitchen where multiple cooks work together to cook and plate a dish by following a recipe. A practical example could be a pizzeria where there are different employees for each station of the pizza's making, one is responsible for mixing and stretching the pizza's dough, another responsible for laying out the ingredients, another one for cooking the pizza and the final one has to pack it and deliver to the client. Each of these workers has a default set of tasks that have to be done for every pizza and a possible set of extra tasks for any special order.

7.3 Future work

Along the whole development period, there were some decision-making given the scope of the work that where influenced by the time and resources available. In the future some of these decisions should be reevaluated and further testing cases should be created.

As a way to strengthen the importance of the Support Interface in this type of systems, more precisely the PSA Production Unit case presented earlier, it would be interesting to perform a more variety of tests that could be able to cover and answer to as many variables as possible.

Unfortunately, due to problems related with Internet and schedule incompatibly between participants and the needed people to run the CPS, it was not possible to realize the Algorithm test as it was initially expected to. The Algorithm Test aims to understand how accurate and precise both the Expertise level Evaluation and Task Evaluation algorithms are with the goal of improving them for future and better usage.

This test would be very importance to the overall function of the system and user experience as it is what marks the presence of a user as part of the system and the better these algorithms are developed, the better it will be able to attribute a valid and accurate expertise level to the worker and this way improve every other aspect related with the system.

Looking at some of the prepositions taken on Section 6.5, it would be interesting to test the CPS with harder, more diverse and more complex tasks to understand how much the Support Interface would be effective with these type of tasks, although it would be expected that the Grades from the CPS without the Interface would not be as high as what was recorded in the executed tests as the tasks would be harder to read and understand. It would also be beneficial to vary the number of tasks per product and the available time on each product and analyze how these factor would reflect on the users performances and once again the expected result would go along the expected before mainly if the number of tasks would increase which according to what the PSA Group of Mangualde also detected on Section 3.2 on Chapter 1.1 and explained through Figure 3.2, a list of tasks gets harder to understand as the number of tasks increases. Finally, and as a way to test the system in a closer environment to the one found on the factory, it would be interesting to test the CPS with participant with more similar characteristics to the actual factory workers.

Apart from these suggested tests, a reevaluation and reformulation of the developed algorithms, that calculates the Expertise level of the workers and convert the objects coordinates into task rating, should take place. Some machine learning algorithms should be studied to look for a better solutions for the evaluation of the humans involved.

After these optimizations of the work developed, the next step should be to add more variables to the equation, extending the monitoring spectrum. By monitoring the stress, anxiety and fatigue levels of the workers, using wearable sensors, as well as other external aspects, such as work and climacteric conditions, that may directly or indirectly influence the operator's job, it would be possible to achieve a more precise evaluation of the tasks and consequently of the expertise level.

7.4 Epilogue

Overall, the elaboration of this dissertation came out as a highly productive research project which allowed for the acquisition of knowledge of multiple concepts within the field of Industry 4.0, Cyber-Physical Systems and Human-Machine Interaction. The study of multiple topics such as smart components, digital twin, interface attraction, contrast and many others offered excellent experience and knowledge on how to research on these type of environments just like the one presented before. It also allowed to understand what steps to take towards the cognitive load reduction using the digital world available reinforcing the interaction between humans and machines.

Bibliography

[1] 123test. *Free psychological tests*. 2019. URL: https://www.123test.com/.

[2] Liliana Antão. "Cooperative Human-Machine Interaction in Industrial Environments". PhD thesis. Faculdade de Engenharia da Universidade do Porto, 2017.

[3] Fair Labor Association. *EXAMINING THE IMPACT OF LONG HOURS ON FACTORY WORKERS*. Sept. 2011. URL: http://www.fairlabor.org/blog/entry/examining-impact-long-hours-factory-workers.

[4] Steven Bradley. *11 Ways To Add Depth To A Design*. Jan. 2012. URL: https://vanseodesign.com/web-design/pictorial-depth-cues/.

[5] Adele Burney. *The Life of a Chinese Factory Worker*. URL: https://woman.thenest.com/life-chinese-factory-worker-9239.html.

[6] Kendra Cherry. *The Control Group in Psychological Experiment*. Nov. 2018. URL: https://www.verywellmind.com/what-is-the-control-group-2794977.

[7] Alan Cooper et al. *About Face. The Essentials of Interaction Design*. 4th edition. wiley.

[8] Belle Beth Cooper. *Novelty and the Brain: Why New Things Make Us Feel So Good*. May 2013. URL: https://lifehacker.com/novelty-and-the-brain-why-new-things-make-us-feel-so-g-508983802.

[9] Stanley Coren and Joel Miller. "Size contrast as a function offigural similarity". In: (1974). URL: https://doi.org/10.3758/BF03203955.

[10] Electron. *Electron*. URL: https://electronjs.org/.

[11] MariaDB Foundation. *About MariaDB*. 2019. URL: https://mariadb.org/about/.

[12] Christian Fuchs et al. "Adaptive Consoles for Supervisory Control of Multiple Unmanned Aerial Vehicles". In: (2013). URL: https://doi.org/10.1007/978-3-642-39330-3_73.

[13] GitHub. *Atom*. URL: https://atom.io/.

[14] IGI Global. *What is Cognitive Overload*. URL: https://www.igi-global.com/dictionary/cognitive-overload/36188.

[15] I. Graessler and A. Poehler. "Integration of a Digital Twin as Human Representation in a Scheduling Procedure of a Cyber-Physical Production System". In: (2017).

[16] PSA Group. *PSA Group*. URL: https://www.groupe-psa.com/en.

[17] PSA Group. *PSA Group*. URL: https://site.groupe-psa.com/mangualde/pt-pt/sobre-nos/.

[18] Danny Halarewich. *Reducing Cognitive Overload For A Better User Experience*. Sept. 2016. URL: https://www.smashingmagazine.com/2016/09/reducing-cognitive-overload-for-a-better-user-experience/.

[19] Ben Hayes. *Is 'flat design' really flat?* Jan. 2013. URL: http://www.benhayes.com/blog/tag/visual-design/.

[20] Dean A Hooper. "Are we designing products for experts or just really good beginners?" In: (2015).

[21] S. Jack Hu. "Evolving Paradigms of Manufacturing: From Mass Production to Mass Customization and Personalization". In: *Proceedings of the Third Conference on Object-Oriented Technologies and Systems* (2013).

[22] IARIA. *The Eighth International Conference on Intelligent Systems and Applications INTELLI 2019*. 2019. URL: https://www.iaria.org/conferences2019/INTELLI19.html.

[23] N. Jazdi. "Cyber Physical Systems in the Context of Industry 4.0". In: (July 2014). URL: http://10.1109/AQTR.2014.6857843.

[24] Jeff Johnson. "Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules". In: (2010).

[25] David Kirsh. "A Few Thoughts on Cognitive Overload". In: *Intellectica* 1.30 (2000), pp. 19–51.

[26] Janie Kliever. *Designing with contrast: 20 tips from a designer*. URL: https://www.canva.com/learn/contrasting-colors/.

[27] Andrew Kucheriavy. *Best Ways to Attract and Hold Online Users' Attention, Part 1*. URL: https://www.intechnic.com/blog/best-ways-to-generate-and-hold-user-attention-online/.

[28] Andrew Kucheriavy. *How to Keep Users Focused and Increase Their Attention Spans, Part 2*. URL: https://www.intechnic.com/blog/how-to-keep-your-audience-focused-and-boost-their-attention-span/.

[29] Peep Laja. *8 Things That Grab and Hold Website Visitor's Attention*. Feb. 2012. URL: https://conversionxl.com/blog/how-to-grab-and-hold-attention/.

[30] Pat Langley. "Machine Learning for Adaptive User Interfaces". In: (1997). URL: https://pdfs.semanticscholar.org/981f/138d7aa144cda97b23b64d12c5f6b9bafcde.pdf.

[31] Gitte Lindgaard et al. "Attention web designers: You have 50 milliseconds to make a good first impression!" In: *Behaviour & Information Technology* 25.2 (2006), pp. 115–126. DOI: 10.1080/01449290500330448.

[32] Ux London. *Users are neither beginners nor experts – designing for the golden mean*. Feb. 2014. URL: https://uxoslo.com/2014/02/17/users-are-neither-beginners-nor-experts-designing-for-the-golden-mean/.

[33] Leonardo Machado. "Towards Interpretable Unbiased Behavioral Pattern Recognition". PhD thesis. Faculdade de Engenharia da Universidade do Porto, 2018.

[34] Nikolaos Makris. "Creating Adaptable and Adaptive User Interface Implementations in Model Driven Developed Software". PhD thesis. The Netherlands: Faculty of Science.

[35] Connie Malamed. *What is Cognitive load?* URL: http://theelearningcoach.com/learning/what-is-cognitive-load/.

[36] Kevin Matz. *Designing Usable Apps. An agile approach to User Experience design*. 978-0-9869109-0-6. Winchelsea Press.

[37] MacDonald Humfrey Automation (MHA) and ExMac Automation. *MacDonald Humfrey And ExMac Create The Assembly Line For A New High Performance Engine At Ricardo*. URL: http://www.exmac.co.uk/macdonald-humfrey-and-exmac-create-the-assembly-line-for-a-new-high-performance-engine-at-ricardo/.

[38] Scott Morris. *Tech 101: What is JavaScript?* 2019. URL: https://skillcrush.com/2012/04/05/javascript/.

[39] Mozilla and individual contributors. *About JavaScript*. 2019. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript.

[40] Luis Neto et al. "A component framework as an enabler for industrial cyber physical systems". In: (May 2018). URL: https://ieeexplore.ieee.org/abstract/document/8387682.

[41] Jakob Nielsen. "F-Shaped Pattern For Reading Web Content (original study)". In: (Apr. 2006).

[42] Oleksandrm. *Contrast In Design*. June 2016. URL: https://www.designcontest.com/blog/contrast-in-design/.

[43] Malgorzata Piernik. *8 ways to reduce cognitive load: Part 1*. July 2017. URL: https://uxplanet.org/8-ways-to-reduce-cognitive-load-part-1-cc2048d1b157.

[44] B Joseph Pine, Bart Victor, and Andrew C Boynton. "Making mass customization work". In: *Harvard business review* 71.5 (1993), pp. 108–11.

[45] Hanna Poffenbarger. "Teaching Tips/Notes". In: (). URL: https://www.nactateachers.org/images/TeachingTips/Teaching_tips_based_on_Cognitive_Load_Theory.pdf.

[46] João Reis, Rui Pinto, and Gil Gonçalves. “Human-Centered Application using Cyber-Physical Production System”. In: (2017).

[47] Santo. *Web Design Articles and Resources*. Mar. 2010. URL: http://www.killersites.com/magazine/2010/fundamentals-of-graphic-design-typography-part-2/.

[48] Mike Sondalini. *Unearth the answers and solve the causes of human error in your company by understanding the hidden truths in human error rate tables*. URL: https://www.lifetime-reliability.com/cms/tutorials/reliability-engineering/human_error_rate_table_insights/.

[49] John Sweller. “Element Interactivity and Intrinsic, Extraneous, and Germane Cognitive Load”. In: *Educational Psychology Review* (Apr. 2010).

[50] Alexandre N. Tuch et al. “The role of visual complexity and prototypicality regarding first impression of websites: Working towards understanding aesthetic judgments”. In: (Aug. 2012). URL: http://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/38315.pdf.

[51] Laurian Vega. *Designing for Novices & Experts, Part 1*. Aug. 2017. URL: https://medium.com/next-century-user-experience/designing-for-novices-experts-part-1-a46bdd5b09a.

[52] Wojciech Zieliński. *How to use contrast in UI Design*. Dec. 2016. URL: https://blog.prototypr.io/how-contrast-works-in-ui-design-21bf75a5a2bf.