POLITECNICO DI TORINO

Faculty of Information Technology Engineering

Master of Science in Computer Engineering

Master Degree Thesis

Digital Twin and HMI for collaborative autonomous mobile robots in flexible logistics

Supervisor:
Prof. Paolo Garza

Candidate:
Claudio Filocamo

Company Supervisor
Concept Reply SPA
Dr. Liudmila Dobriakova

Academic year 2019-2020

Alla mia famiglia
† A mio zio Gino

Contents

List of Tables

List of Figures

Nomenclature

Acronym	**Meaning**
DT	Digital Twin
HMI	Human Machine Interface
AGV	Automated Guided Vehicle
WMS	Warehouse Management System
MES	Manufacturing Execution System
AI	Artificial Intelligence
ML	Machine Learning
IoT	Internet of Things
IIoT	Industrial Internet of Things
UC	Use Case
Cobot	Collaborative Robot
SSE	Server Sent Events
MMI	Man Machine Interface
OT	Operator Terminal
GUI	Graphical User Interface
RFID	Radio-Frequency Identification
PLC	Programming Logic Controller
OPC UA	Open Platform Communications Unified Architecture
DBMS	DataBase Management System
DDM	Data-driven model

Introduction

The digital transformation is pervading our society, boosted by the development of new technologies and a lot of companies are undergoing radical changes, which apply both to products and processes, in order to comply with the requirements that the competition imposes.

According to Istat, in Italy, 97% of companies with more than 500 employees are using digitalization technologies or are investing in them.

Digitalization is just a piece of the puzzle of industry 4.0.
The "fourth industrial revolution" (Industry 4.0) broadly describes a digital manufacturing environment that combines advanced manufacturing techniques with the IoT to create not only an interconnected manufacturing enterprise but one that communicates, analyzes, and uses information to drive further intelligent action back in the physical world [10].
Automation comes together with digitalization, modularity, networks, and connectivity in the automation context enabling the next generation of collaborative production networks to deal with a very dynamic market, increasing productivity, speed, efficiency and quality. Cloud computing, IIoT (Industrial Internet of Things), Big Data, Artificial Intelligence (AI) are the mainstays today's smart industry is based on.
The concept of industry 4.0 (Smart Factory) can be explained by means of:

- Smart production: new production technologies that enable the collaboration between all the elements involved in the production stage: operators, machines and instruments.
- Smart services: all the IT infrastructures and the techniques that allow the integration of systems, but above all the structures that let companies integrate between each other.
- Smart energy: very efficient systems that reduce energy waste according to the typical paradigms of sustainable energy.

In this context, the concept of digital twin is gaining more and more importance, in fact, many organizations already exploit it or are willing to.
So, what is a digital twin?
It is an evolving way of simulating the historical and current behavior of a physical machine or a production process as an exact replica of the model it reproduces, since it is based on the same data, functionalities and communication interfaces, with the aim of optimizing it, validating it, monitoring it and hence preventing (taking actions against it) or reducing the risk of breakages, problems or errors that could occur.
The reason the usage of Digital Twin is becoming wide spread nowadays is that those technologies it relies upon, have become much more mature than they have been in the past and they include low-cost reliable data storage and computing power, the availability of robust, high-speed wired and wireless networks and cheap, reliable sensors.

As a way of interacting with the simulated machine, process or system, which can vary a lot and evolve over time, it seems natural to think about HMI, acronym for Human Machine Interface. An HMI is, indeed, a layer between humans and the system making them communicate in a simple, efficient and secure way and it comes with different characteristics according to the outcome of the analysis which should take into account parameters like cognitive load, technologies needed and other factors deemed important by the company. Moreover, by means of an HMI it is possible to visualize meaningful data, always keeping the system under control.

This thesis work has been drawn up during a collaboration for a project, whose specialization area is the smart factory, with Reply S.p.A.
The title of this MIUR Founded project is ICOSAF (Integrated COllaborative Systems for smArt Factory) and its leader is CRF (Centro Ricerche Fiat). In particular, its aim is to develop and integrate technologies for collaborative workspaces following the principles of industry 4.0 (interconnected automation) and industry 5.0, which is about humanization and re-use of resources.
The leading actors of industry 5.0 will be cobots rather than robots, because they are able to work together both with humans and other robots in shared areas, in a secure and efficient way.
Operator enhancement is one of the cores of ICOSAF, in fact, workers cooperate with cobots and are assisted in low added value tasks in order to exploit their intelligence and flexibility in tasks where the human factor is more important, being able also to personalize the final products.

AGVs (Automated Guided Vehicles) are examples of collaborative robots, since they are smart, autonomous and capable of coping with coordinated operations.
AGV is a pilotless transport system able to rationalize transport between processes

and provide parts to production lines.
In modern factories, the necessity of increasing production volumes, and at the same time, reducing costs in order to be more competitive in the global market, is more and more important.
This need can be partially satisfied automating the production and the logistic process thanks to the help of robotics.
Nowadays robotic production cells exist and innovative automated solutions in many industrial production plants are integrated with them.
AGVs have been introduced in the fifties in order to improve goods transport. Thanks to their efficiency and flexibility, many systems based on AGVs are used in production lines and at the end of them, precisely in the so called automated warehouses, where enormous quantities of raw materials are moved every day.
Hence, due to the fact that AGVs are pilotless, beyond moving big loads, they need to orient and move in the production plant in a precise and secure way.

Regarding the project addressed, the Use Cases [18] proposed by the project partners are reported in table 1.

The activity which has been conducted is about the development of a digital twin in a manufacturing environment and the development of an HMI interacting with it, following the principles just presented.
In particular, the work will focus on Use Case C both from the simulation point of view and from the User Interface point of view.

ID	**Title**	**Partner**	**Headquarter**	**Brief Description**
UC-A	*"Postazione di assemblaggio alberi controrotanti"*	CRF	CRF Melfi	*"Postazione con robot collaborativi integrata su due linee basate su AGV in sostituzione dei conveyor"*
UC-B	*"Assemblaggio collaborativo di inserti su scocca CFC"*	Adler	Adler Airola	*"Montaggio e incollaggio di inserti metallici sulla scocca in carbonio"*
UC-C	*"AMR (Autonomous Mobile Robots) collaborativi per logistica flessibile"*	CRF	CRF Melfi	*"Postazione di magazzino logistico in-plant ottimizzato per automazione flessibile."*
UC-D	*"Qualità Proattiva su punti di saldatura RSW tramite analisi "Big Data" e postazioni collaborative di controllo"*	CRF	CRF Melfi	*"Sistema di controllo della saldatura a punti basato su una prima selezione dei punti a "rischio" basato su un'analisi con approccio "Big Data" dei dati dalle pinze di saldatura unito con un dispositivo di controllo automatico-collaborativo della qualità dei punti di RSW."*

Table 1: Use Cases

Part I

First Part

Chapter 1

Purpose

The objective this thesis work in this project phase is about the integration between an HMI and a Digital Twin to simulate, optimize and predict what will happen in a collaborative manufacturing work area.
According to the Industry 4.0 statements, digitalizing existing processes is a widely used technique to allow companies to have a complete digital footprint from design until the end of the product life cycle. Moreover, since the physical implementation of the Use Case is still being developed, digitalizing it helps in design, implementation and integration of both the HMI and communication system. It emerges the need for a virtual system that simulates a physical one, receives real-time simulated data that must be the exact replica of what will be actually manipulated when it will be physically implemented (such as work orders from the MES or logistics orders from the WMS).
Hence, the first step will be the virtualization of a system that will be controlled by an HMI (to be developed, too) and that will provide information in the other direction, meaning from the simulation to the HMI, to be visualized in a usable UI. The Digital Twin will have specified parameters so that it can perfectly fit the required demands and that will be provided by the project partners and by CRF of course. Some of them include the speed of Kuka, the speed of the operator (statistically calculated from real workers and adapted in the digital prototype), the picking time of Kuka, the picking time of the operator, the number of shelves, the number of cells present in each shelf, the speed of the AGV, the positions of the actors involved, their behavior in case of error and so on.
Some of them can be modified by the user who is launching the simulation to have different simulation parameters to test and also to see what would happen in case something would be slightly different with respect to the standard behavior expected, such as the picking delay both for Kuka and for the operator. The Digital Twin will not be used just to have a system to integrate with the HMI in order to

test it, but it will be used also to generate interesting insights, make predictions and optimize the processes that will be developed.
Once the Digital Twin production stage will be terminated, there will be the choice of the devices to be used by the remote operator and by the operator in the shop floor by means of an accurate analysis which takes into account technical parameters, usability and other factors.
This will be the beginning of the HMI development phase. As explained before, the human machine interface should permit a bidirectional interfacing going from the simulation to the user interface (there is a way to let the remote operator choose what should happen in the simulation in case of errors) and vice versa (there are controls on the simulation to generate an error and this will be visible on the UI).

Chapter 2

Digital Twin

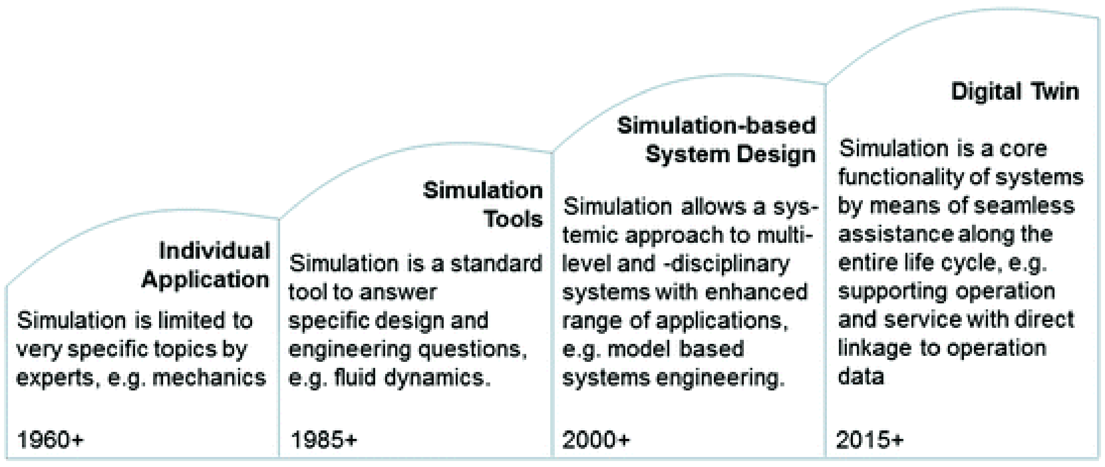

Individual Application
Simulation is limited to very specific topics by experts, e.g. mechanics
1960+
Simulation Tools
Simulation is a standard tool to answer specific design and engineering questions, e.g. fluid dynamics.
1985+
Simulation-based System Design
Simulation allows a systemic approach to multi-level and -disciplinary systems with enhanced range of applications, e.g. model based systems engineering.
2000+
Digital Twin
Simulation is a core functionality of systems by means of seamless assistance along the entire life cycle, e.g. supporting operation and service with direct linkage to operation data
2015+

Figure 2.1: Digital Twin Evolution [8]

By taking a look at the picture above, it is possible to see the evolution of the concept of digital twin throughout the years.

The concept of simulation can be traced back to the sixties, when NASA used simulated environments for space programming.
Simulation, in fact, was used to reproduce critical situations that could have happened once the astronauts had left. Moreover a sort of twin was used during the ruinous space journey Apollo 13 to get back the spacemen safe and sound.

In the eighties simulation tools were used to try to answer to specific problems in order to find a possible solution to them.

At the beginning of the new millennium, model based systems started to become wider in terms of range of applications, but still it was not one of the core features of system development.

It is between the 2014 and 2015 that the concept of Digital Twin finally had a chance to become a core functionality for companies, integrating physical systems and virtual ones by means of the exchange of data and information along the entire lifecycle of the product.

The cost for computation and model-building has decreased a lot and this, of course, gave a boost to the development of digital twins, but there are still barriers to their widespread implementation.

Important examples include the need for data and for a suitable development environment [3].

2.1 Objectives

A Digital Twin comes with two main objectives (depending also on the state of the project when it is employed).

- Experimental: The Digital Twin technology is exploited in a pre-project stage. It perfectly reproduces the properties of a real system, but the costs to develop and test it are much less with respect to the physical system. Moreover, the results obtained by the digital prototype can be used to improve the real system.
  Industries find it very useful to have cost saving and to reduce the Time-To-Market.
- Predictive: Production processes need to be performing, efficient, controlled and fast. The Digital Twin technology can be exploited also to generate forecasts about possible errors or wastes. This leads to higher performances and profit margin.

According to [13], the concept of Digital Twin has a great potential also for:

- Continuous Improvement: The Digital Twin nature allows to involve and integrate many other technologies and digital models which can evolve and configure themselves as key factors for further business improvements.
- Scenario Analysis (What-if): By means of user interfaces more and more advanced, it is easy to make use of widespread functionalities such as Predictive Analysis, Machine Learning for instance, in order to evaluate different mode of operation scenarios of the system, to recognize the best actions to be then applied on the physical system.
- Visibility: A DT enables the visibility related to the functioning of machines as well as larger interconnected systems.
- Insight: Digital Twin can be exploited as a means of communication and documentation in order to gain further information about one or more systems.
- System Integration: When correctly designed, a Digital Twin can be connected to other business applications in order to extract useful data for their own operation (or to communicate outcomes related to it).
- Disruptive Business Models: Not only this technology supports and pushes the traditional business models, but expands them enabling new ones (such as data-driven or PaaS business models).

2.2 Digital Twin principles

Digital Twin is based on three main pillars [13], made possible by the recent technological improvements: connectivity, digitalization, intelligence.

2.2.1 Connectivity

- Sensors: sensors are able to acquire signals that let the Digital Twin acquire operational and environmental data of the context they are in.
  The price of standard sensors has decreased a lot due to the emerging of MEMS (Micro Electro Mechanical Systems) technology.

- IoT: sensors communicate data to the virtualized environment by means of the integration technology (which include edge computing, standard communication and security interfaces) between the physical world and the digital one, and vice versa.

- Big Data: the data that has been acquired by sensors (operational and environmental) are aggregated and combined with other company data.

2.2.2 Digitalization

- Simulation modeling: the digital side of the Digital Twin is an application that combines the components previously described in a digital model, aligned almost in real time with the systems it reproduces.
  The technologies used to realize these models strongly depend on the nature of the systems themselves: while a mechanical component can be easily simulated exploiting techniques based on physics and mathematics, a more complex system, as an assembly line can be modeled using agent-based-modeling or discrete-event modeling techniques.

2.2.3 Intelligence

- AI: Artificial Intelligence is able to make machines learn from experience, correcting the behavior when something does not act like expected and therefore making them more "human". Thanks to the the availability of large datasets, the capability of processing them and the more advanced techniques of machine learning, such as reinforcement learning, AI can actually work perceiving and analyzing different situations, finding the best decisions in order to achieve the required purpose. Moreover, AI can be applied at different stages of the DT development.

- Analytics: analysis techniques can be exploited to study more in details the collected data by the sensors in order to discover some patterns in them that can be eventually used for decision making, gaining insights about the causes that actually influence the business.

- Actuators: In case the AI is in charge of defining the actions related to a physical system, the actuators could be triggered to produce a movement, but in general the outcomes could be more strategic, meaning business decisions that could change processes already running for example.

2.3 Types of Digital Twin

Virtual copies of many products are being widely used in the manufacturing phase. Several types of simulators exist today [11].

- Product Twins: Producers often use a virtual prototype of a product before setting up a production line, in order to make some prediction based on the analysis of how it will perform under various conditions also discovering what issues may occur.This procedure enables them to make necessary adjustments and to have a more efficient design and production.
  As a result, this kind of technology helps to reduce production expenses and time-to-market, while improving quality.
  Afterwards, product twins can be exploited to monitor product performance in the physical world.

- Process Twins: These models are used to simulate manufacturing processes mainly. A virtually simulated production process can show different scenarios and what would happen in different situations.
  This is used by enterprises to enhance the production methodology, improving efficiency. The designed process can be further optimized and integrated with product twins that can simulate each equipment involved. Thus, a company can perform preventative maintenance, avoiding costly downtimes. Manufacturing operations, beside being more efficient, will eventually be safer and faster.

- System Twins: Generalizing the models already presented, it is possible to model an entire system (such as a plant of a factory).
  These kind of models collect huge amounts of operational data produced by sensors, devices and products involved in the system itself, gain important insights and create new business opportunities in order to monitor and optimize all the simulated processes.
  Experts assert that in the near future, the Digital Twin technology will pervade our society, replacing human workers in charge of diagnosing equipment malfunctions. This is already happening in some companies, because DTs are very effective, since they gather information from sensors in the online mode and are able to discover serious damages making their substitution of fix easier.

Taking it to the extreme, some futurist affirm that digital copies of humans will be produced, too.
These twins will be able to anticipate the behavior of the corresponding simulated humans, making decisions on their behalf.

2.4 Digital Twin examples

2.4.1 General Electric's Turbines

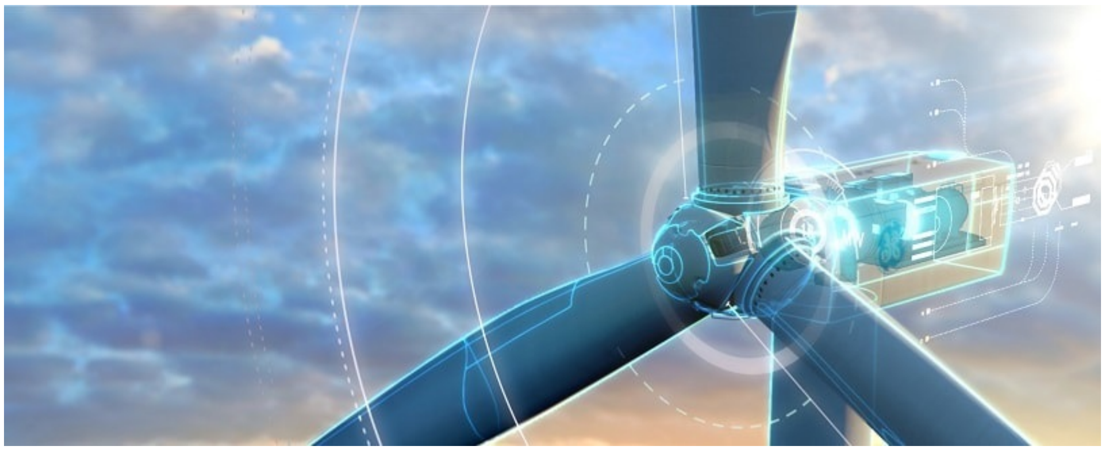


Figure 2.2: Turbine DT [12]

GE, an american multinational corporation founded in 1892, has always been an innovative company from the technology point of view. To be more specific, this company applies to its turbines, that are being used both in wind power plants and inside jets motors, a dense network of sensors that communicate with a virtual Digital Twin almost in real time. From the moment a real turbine is sold, this informs its virtual copy about its usage and this permits not only to register, acquire and analyze data in a continuous way, but also to detect damages, differences, decay forecasting and issues resolutions already during the testing phase.

2.4.2 Tesla Cars

As for GE, Tesla has exploited the technological evolution and it became a cornerstone point for its company growing strategy.
The cars being sold, indeed, transmit their virtual twins every information that could possibly be useful for the company itself in order to improve the efficiency of its products, but not only.
According to Roberto Saracco [26], Head of industrial doctoral school of Eit Digital, Tesla everyday receives the equivalent of over 2 millions of kilometers tracks, obtained by summing up the contributions of each car. This enormous amount of data allows them to build a map which is always updated about roads and to verify the presence of structural issues, meaning those which depend on design issues.

2.5 Data

Data is probably the most important part in the development of a Digital Twin. First of all data needs to be acquired throughout its operational life, so that the model represented is always maintained up-to-date.

Data must be, of course, relevant and should reflect the real behavior of the simulated system to feed the virtual one.
Indeed, data must describe the environment surrounding the Digital Twin, but also its evolving state and how it works in these circumstances.

As stated by [3], moreover, collecting data just once is rarely enough. It must be timely and in agreement with the physical system it is representing.

This is why digital twins are so closely related to the IoT, with connected devices reporting operational data to the model for processing and analytics.

Since the cost of computing and communication continues to decrease, the IoT will become increasingly widespread and provide the necessary data flows more easily.

If the data collected were not meaningful, it would be impossible or it would lead to wrong insights when trying to produce forecasting.

In order to provide the Digital Twin some data to be processed, it is meaningful to think about how this can happen.
When facing different projects, these may require different ways of data interaction and a tool that is able to identify, handle and maybe also process raw data can be a key point for DT development, helping to simplify its creation.

Besides, understanding the external data formats and query standards the DT will have to interact with is a fundamental step.
Hence, the tool to be chosen needs to take care of native access to a range of data management systems that could be very wide.

Once developed, the Digital Twin should be able to integrate and communicate with a wide range of data sources, such as databases or parts of an enterprise's IT infrastructure as they are developed and deployed.

This leads to the so called data-driven approach which is a way for business decision making, such as product data, on the basis of the collected and analyzed data.

As explained by [13] data driven modeling provides the so called "digital" side of the Digital Twin.
On the contrary of simulation models, which receive and process data from a physical twin, data driven models (DDMs) consider also mathematical and statistical knowledge to study which relationships exist between the outcomes obtained with the given inputs.

New technologies such as advanced sensors and data collection and analysis techniques turned out to be a boost for DDMs, which are gaining more and more attention beside simulation ones.

2.6 Modeling Environment

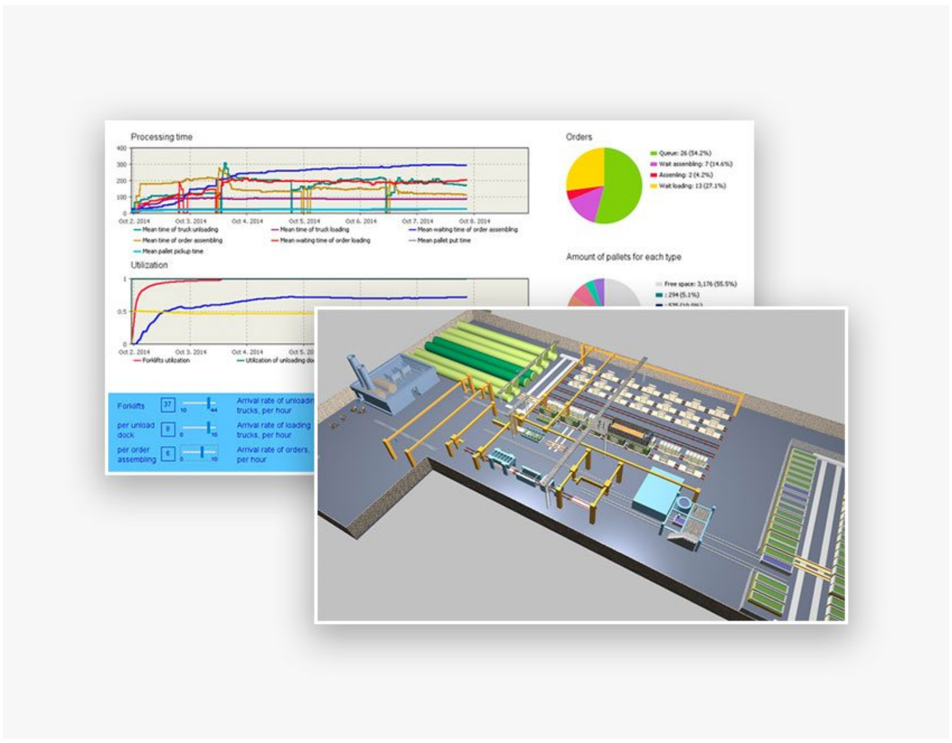


Figure 2.3: Modeling Environment [27]

According to [3], developing a model of a real-world system very often involves multiple components and therefore can quickly become complex.

Models can also expand into multiple realms. For example, thinking about a supply chain analysis, this could benefit from modeling other contexts such as warehouse and retail operations.

However, modeling environments as the ones described before may require different approaches, possibly due to the information available, or the nature of the operations.
Warehouse operations, for instance, are usually modeled differently to network-level supply chain operations.
Linking and integrating complex and disparate processes and operations requires a flexible modeling environment, ideally one able to connect differing modeling methods .

Another aspect that needs to be taken into account is software dependency.
The aim indeed is to minimize the number of software platforms used for the creation and operation of a digital twin, hence helping streamline support, maintenance, and further development.

Digital twins are, in a large part, simulations that need to represent some aspects of the real world in some form, and a capable modeling environment is a fundamental key.

That being said, when considering a digital twin platform, attention should go beyond the ability to create a good model.
Developers should consider also information flows as well as visualization. This will be done in further chapters in order to distinguish and choose among the considered tools.
Developing a digital copy requires capturing the necessary real-world complexities, and this quite often demands more than one modeling method.
A multimethod modeling environment can of course simplify a lot the development, because if a single tool could accurately capture all the asked details, this would be much crucial.
The speed of deployment and the digital twin's lifetime support will also benefit from using one tool only.

2.7 Architecture

A digital twin can be exploited to accomplish different goals.
It can collect data from the real world exploiting sensors and their environment, thus reflecting its status in a digital domain.
The state is saved and can be further analyzed by means of an artificial intelligence that can apply machine learning algorithms on the data stored or some prediction scenario can be developed starting from the models obtained.
At the same time if the data manipulated by the digital twin is elaborated in real time it could be possible to provide an alerting system for a remote operator for example, building up a typical control scenario.

All the functionalities that generally can be found in a digital twin are represented in the figure below.

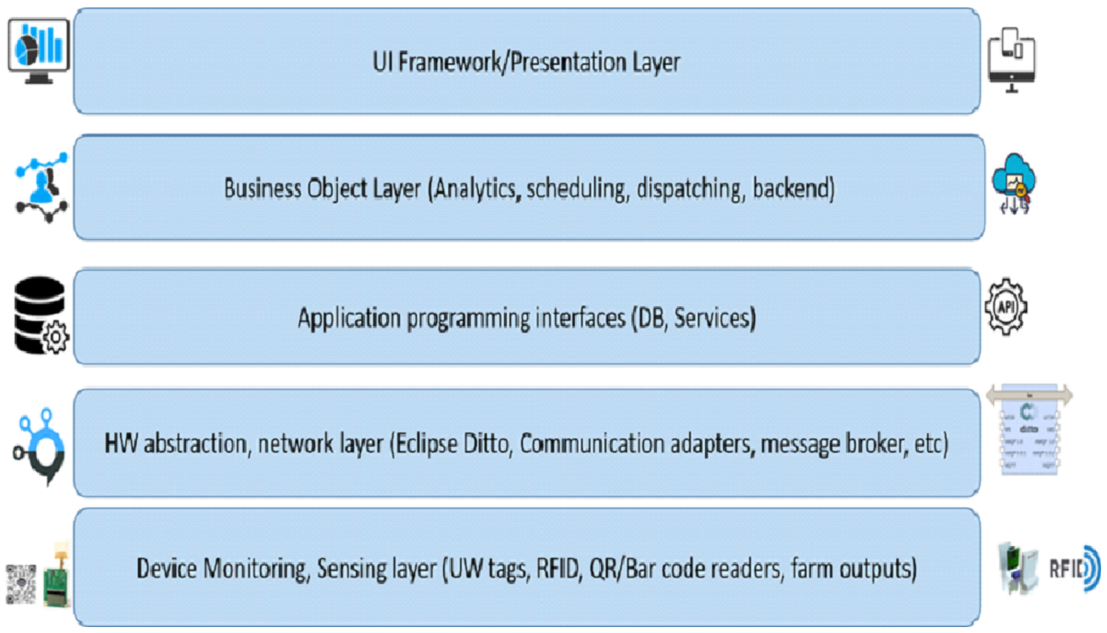

UI Framework/Presentation Layer
Business Object Layer (Analytics, scheduling, dispatching, backend)
Application programming interfaces (DB, Services)
HW abstraction, network layer (Eclipse Ditto, Communication adapters, message broker, etc)
Device Monitoring, Sensing layer (UW tags, RFID, QR/Bar code readers, farm outputs)
API
RFID

Figure 2.4: Digital Twin Architecture

So, the main functional components can be described as layers with different meanings and services offered.
The first layer is the sensing one and it represents all the physical sensors used by the Digital Twin, such as RFID (Radio-Frequency IDentification) tags or UWB, used for precise real time location of the goods or the AGVs, inside the warehouse or inside the manufacturing plant, or even QR/Bar code readers.

The second layer is the data abstraction and communication one and it is responsible for making transparent the origin of the data source for the system, providing bridges or gateways. Raw data coming from sensors are translated to be usable, saved, analyzed and possibly visualized.
Application programming interfaces is about data persistence in order to allow further data elaboration, visualization and analysis. Data can be saved in relational databases (SQL) or non-relational ones (noSQL). In this layer data is stored and the way in which this should happen is defined.
The Business Analytics Layer focuses on data simulation and prediction applications. Task scheduling and potential interfacing with company proprietary systems (like MES) happens here. In case the stage being faced is pre-projectual and there is no data coming from an actual MES that could be exploited yet, also MES data is simulated and a prediction about how its outcome would be, is produced in order to exploit it for further developments.
On top of this stack there is the presentation layer which is about the way data will be presented to the final user to be clear, understandable and usable. This is the HMI layer.

Data is often acquired by means of HTTP/IP communication protocol and then the abstraction layer is exploited to gateway data. Then, data management services are notified about the arrival of data and can be finally used in any way it was intended to (for example presenting it with Angular, HTTP, nodejs and so on), but only registered sensors can write this kind of information. Apart from the layers just presented also the security and the human resources aspects need to be considered as prerequisites for a proper Digital Twin development.
The duties as well as the skills required to develop and maintain a DT, in fact, demand an adequate knowledge of the people in charge of interacting with it, but on the other side it can represent for the employees a great opportunity for professional growth, becoming actors of the digital transformation.
Nevertheless guaranteeing the security of data management systems is another key aspect in order to avoid losses or damages.

2.7.1 Hardware components

Being an IoT system, the hardware utilized to build a Digital Twin includes:

- Devices for a remote dashboard access
- Devices for control
- Servers
- Routing or bridge device
- Sensors

Each of these devices is involved in a key task such as system activation, action specifications, security, communication and detection to support specific goals and actions.

The first ones are used by the remote operator to access the dashboard in order to visualize data, while the second ones are devices used to control data and interacting with it.
Often these two tasks are incorporated in one device which is used by the remote worker to both visualize and control the data presented. PCs or Tablets are generally the best choice for this kind of devices.

Servers are used to store data. As for ICOSAF, all the relevant data is deployed to the Reply Cloud. The software simulation tool communicates with the services (that permit to access to the relational data, stored in a SQL database), written in nodeJS, exposed by the cloud as RESTful APIs.

Connectivity is one of the pillars of the Digital Twin technology, as real time moving objects which can be located also in remote areas need to communicate with each other and to be surrounded by a smart environment.

Sensors are implicitly simulated by the software simulation tool. The framework that will be used (AnyLogic), in fact, contains the logic able to let robots perform different tasks. Therefore there is no need for real sensors in a virtually simulated environments, but they are simulated by the behavior itself of AnyLogic actors. In fact, they are able to correctly position to stop points, pick the specified good from a shelf, avoid obstacles and perform all those tasks that would require a physical sensor or a machine vision camera to identify hindrances.

2.8 State of the art - Simulation Software Tools

Digital twins are significantly more accessible than before, and their careful use can cut costs and drive profitability. They are already being deployed in a wide variety of industries, and companies who fail to take advantage are at risk of being left behind.
A powerful and flexible simulation environment is key to the successful development of a digital twin. It is necessary for a digital twin to be able to interact with a variety of other software systems, data sources, and users.
Furthermore, the scope and scale necessary to model whole real-world objects, operations, and processes through their entire lifecycles means complexity must be easily manageable. [4]
So, which aspects need to be considered when choosing a software tool to develop a Digital Twin?
According to [5] the comparison should be made taking into account:

- Vendor and Markets
- Technical Compatibility
- Model Building
- Animation
- Support / Training
- Other information (such as software costs, new features and vendor comments)

The software tools under analysis are mainly: AnyLogic, Plant Simulation and Matlab.
These tools are the ones that have been taken into consideration by the project partner in charge of developing the Digital Twin.

Vendor and markets is about the vendor itself, the typical applications for which the software tool is generally used, the primary markets it is used and the vendor's other softwares.
As for the technical part, compatibility is one of the most important challenges for any software being exploited.
In particular, the supported operating systems, the compatible softwares to perform specialized functions, the capability of being controlled or run by an external program and the multiprocessor CPU support are the main aspects in this case.
When building a model it is also important to have in mind that it could be useful to have run time debug, optimize the code, develop the model using programming,

mix discrete and continuous modeling and so on. This is what model building is about.
Animation is probably the most impressive part of the simulation. Indeed 3D animation, real time viewing, cad drawing imports are perceived as charming features which allow to see at a glance what is going on in the simulated process.
Support / Training takes into account the documentation which explains the usage of the features offered by the tool, but also the training courses (including on-site training), discussion areas and user support.

Tables 2.1, 2.2, 2.3, 2.4, 2.6 point out the main differences between AnyLogic and Plant Simulation.

Software Tool	**Vendor**	**Typical Applications**	**Primary Markets**
AnyLogic	*AnyLogic North America*	*Multimethod general purpose simulation tool. Discrete event, agent-based, and system dynamics modeling.*	• *Supply Chains* • *Transportation* • *Warehouseoperations* • *Rail logistics* • *Mining* • *Oil and gas* • *Road traffic* • *Passenger flows* • *Manufacturingand material handling* • *Healthcare* • *Business processes* • *Asset management* • *Marketing* • *Social processes* • *Defense*
Plant Simulation	*Siemens Product Lifecycle Management Software Inc.*	*Discrete-event simulation, visualization, analysis andoptimization of productionthroughput, material flow, andlogistics*	• *Automotive OEM and supplier* • *Aerospace and defense* • *Consumer products* • *Logistics* • *Electronics* • *Machinery* • *Healthcare* • *Consulting*

Table 2.1: Vendor and Markets comparison [5]

First of all, AnyLogic is an open source software, while Plant Simulation is a Siemens proprietary one. By taking a look at table 2.1, we can also see that AnyLogic is more widespread in terms of primary markets with respect to the competitor under analysis.

Software Tool	**Supported Operating Systems**	**Compatible Software to Perform Specialized Functions**	**Being Controlled or Run by an external program**	**Multi processor CPU support**
AnyLogic	*Windows, MAC, Linux*	• *Excel, Access and any DB* • *OptQuest* • *Stat::Fit* • *Any Java / DLL library e.g. for bayesian or neural networks.*	*AnyLogic models can be exported as standalone Java applications that can be run from/by any other application. They could be also run online via AnyLogic Cloud web service.*	*YES*
Plant Simulation	*Windows*	• *Matlab* • *Excel* • *SAP* • *Simatic IT* • *Teamcenter* • *Autocad* • *Microstation*	• *Parameterizing fromMS Excel* • *Siemens PLCSIM Advanced* • *OPC, OPC UA, ODBC* • *MS Windows* • *Oracle*	*YES*

Table 2.2: Technical Compatibility comparison [5]

Moreover, table 2.2 which is about the technical compatibility, shows that AnyLogic's supported operating systems are Windows, MAC and Linux.
Plant Simulation is only available for Windows machines.

One of the most important features offered by AnyLogic is undoubtedly the possibility of writing custom Java code to implement the logic. This software tool, in fact, is strictly based upon Java libraries.
Both software tools support multi processor CPU.

Software Tool	**Input distribution fitting**	**Graphical model construction**	**Optimization**	**Run time debug**	**Model building using Programming / Access to programmed modules**	**Mixed discrete / continuous modeling**
AnyLogic	*31 predefined distributions and custom distributions. Stat::Fit, ExpertFit, and other software for distribution fitting*	*YES*	*OptQuest is included, additionally users can employ any custom optimization algorithm*	*YES*	*YES*	*YES*
Plant Simulation	*22 predefined distributions*	*YES*	*Genetic Algorithm, LayoutOptimizer, Neural networks, Hill Climbing, Dynamic Programming, Branch and Bound*	*YES*	*YES*	*YES*

Table 2.3: Model Building comparison [5]

As for the model building features, both AnyLogic and Plant Simulation support mixed discrete and continuous modeling, graphical model construction and model building using programming (or access to programmed modules). The last feature mentioned is a key one, when building a model, because it allows the developer to add some specific behaviors to the simulated system.
With regards to the input distribution fitting, AnyLogic has 31 predefined distributions and custom ones, whilst Plant Simulation has 22 predefined ones.

Software Tool	**Animation**	**Real-Time Viewing**	**3D Animation**	**CAD Drawing Imports**
AnyLogic	*YES*	*YES*	*YES*	*YES*
Plant Simulation	*YES*	*YES*	*YES*	*YES*

Table 2.4: Animation[5]

Software Tool	**Consulting Available**	**User Group or Discussion Area**	**Training courses**	**On-site training**
AnyLogic	*YES*	*YES*	*YES*	*YES*
Plant Simulation	*YES*	*YES*	*YES*	*YES*

Table 2.5: Support/Training[5]

Tables 2.4 and 2.5 show that both the software tools under analysis embrace the same considered features.

Software Tool	**Student Version**	**Major New Features (since 2015)**
AnyLogic	*Free AnyLogic Personal Learning Edition*	• AnyLogic Cloud, a web service forsharing models and running themonline on any device. • The Road Traffic Library for detailed modeling of vehicle movement on roads. • The Material Handling Library for thesimulation of manufacturing systems andoperations.
Plant Simulation	*Free Version Available*	• Enhanced worker, robot, mixer, motionpaths and visualization. • New Simtalk, OPC UA and SiemensPLCSIM Advanced connections.

Table 2.6: Other Information[5]

In figure 2.6, other important information are represented.
AnyLogic offers the possibility to connect and upload the developed model to the AnyLogic Cloud, where it is possible to run the model online on any device equipped with a web browser.
Plant Simulation on the other hand offers support for OPC UA (the model can be controlled or run by means of it), which could be very useful especially for smart factories.

What about Matlab, instead?
According to [9], starting from the price, once again AnyLogic has a free professional and learning version while Matlab does not have a free version and its cost is about two hundred dollars/year per user.

Matlab is a programming, modeling and simulation platform that allows users to analyze data, create algorithms, build models and run deployed ones, while no communication with external services is available: it is a closed loop software.
As for the features and functionalities as well as the ease of use, it is important to know how to code in Matlab, and this could be quite complex to learn.

AnyLogic, on the other hand, is much more intuitive and straightforward.
The product features offered by AnyLogic result to be wider in terms of 3D model building, in fact Matlab does not allow 3D modeling, agent-based modeling, continuous modeling and discrete event modeling.
Both AnyLogic and Matlab provide graphical modeling and design analysis and are available for Windows, MAC and Linux.

For the purpose of this thesis work, according to the brief analysis conducted, AnyLogic was chosen mainly for pricing issues, coding issues (Java was preferred to

any other language offered by the other tools) and modeling ones, in fact, AnyLogic is the only tool that allows Agent-Based Modeling, Discrete Event Modeling and Dynamic Modeling and moreover offers the possibility to call external APIs which will be a key point for the development of the requested Digital Twin.
Moreover, AnyLogic is better with respect to Matlab/Simulink from the educational and presentation purposes, while for pure controlling the latter could have been better.
At this point, it is useful to have a closer look at the chosen software:
AnyLogic is already in use in a lot of business fields, successfully delivering the advantages digital twins offer. The AnyLogic simulation modeling environment is the best one for quick development without compromise – as demonstrated by world leading organizations [3].
AnyLogic is the leading simulation modeling software for business applications, used worldwide by over 40% of Fortune 100 companies. AnyLogic simulation models permits analysts, engineers, and managers to gain deeper insights and optimize complex systems and processes across a wide range of industries [4]. Furthermore, it comes with predefined libraries for simulating processes (Process Modeling Library), people flows (Pedestrian Library) and also railway related ones (Rail Library).
It is possible to generate state charts, action charts, flowcharts, stock & flow diagram and the modeling language also includes graphical elements, tools to connect to external sources of data, predefined 3D actors, and much more.
Being extendable with Java code, it is possible to develop complete virtual systems from data acquisition to data manipulation, visualization and post-processing.
There is no specific requirement to use this software tool (it is freely downloadable directly from the AnyLogic website), apart from knowing how to develop code in Java, but it is needed to be aware of what needs to be simulated in order to choose how to implement the requested functionalities, including choosing the data source to exploit. AnyLogic, in fact, developed a built-in database feature to help in building models that require a certain amount of external data for initialization and work. This leads to four primary benefits [6]:

1. It removes extra connectivity layer between Excel or the external DB and model.

2. It is a useful instrument for visualization of agent type hierarchy and development of the data model.

3. It may act as an information dealer for agents, simplifying the algorithms, encapsulating the data and serving as a unified data source for parametrized variables.

4. It stores the simulation statistics, simplifying the data post-processing and export.

2.9 Simulation goal

The use case that will be simulated is Use Case C.
The Use Case C purpose is to implement a collaborative workstation within the logistic space of a production establishment with the aim of improving the productivity and the competitivity of the area.
Exploiting a cobot and creating promiscuous environments where humans and robots may cooperate, the operations will be much more efficient in order to drastically reduce the low value added operations, such as components movements, kit preparation and so on, in favor of a more effective employment of workers.

The introduction of smart and automatic systems, like collaborative robots, remarkably decreases the number of errors that humans can possibly do in the act of picking the correct item in the exact quantity from the shelves for example.
Moreover, in order to reduce the number of possible human errors a pick-to-light system, enlighting the items to be picked, will be used.

The architecture conceived for realizing this use case is represented in the following image.

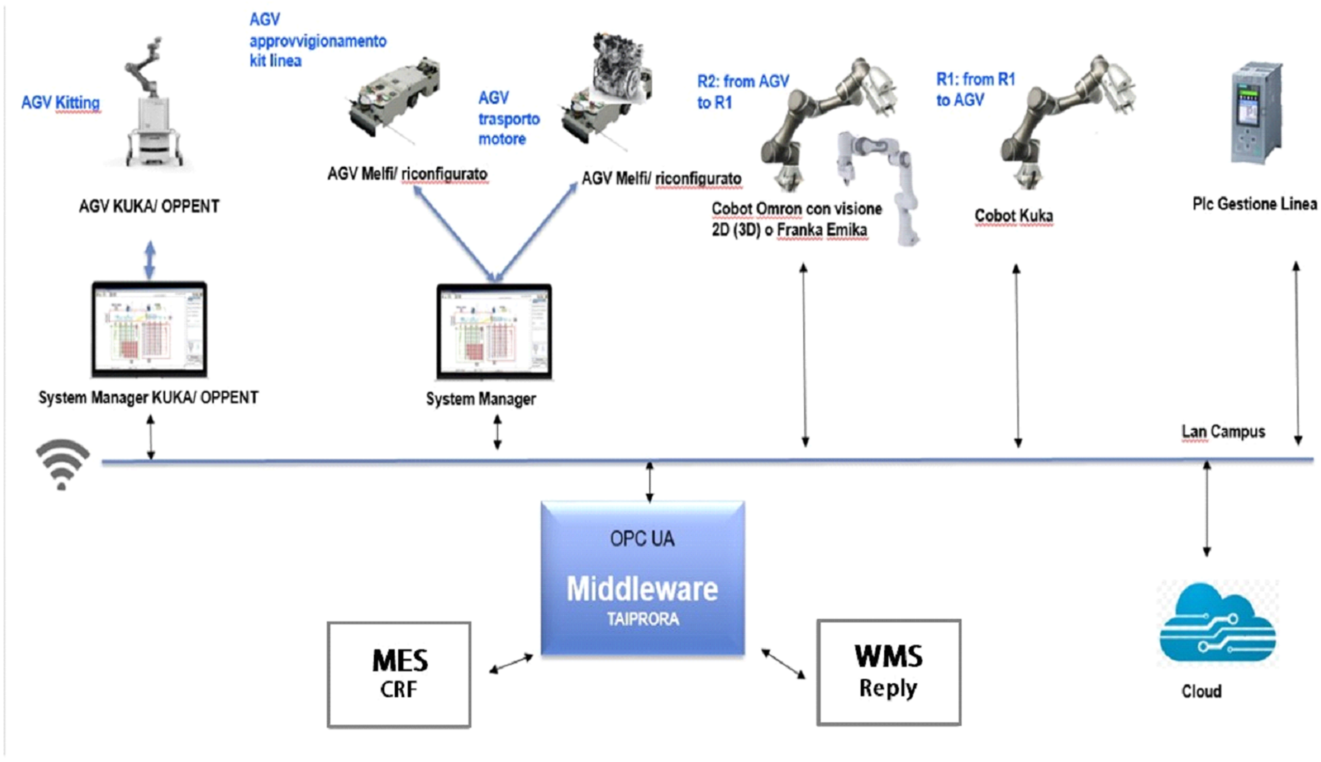


Figure 2.5: Use Case C Architecture

This (2.5) is one of the first sketches of the Use Case C architecture.

Initially, it was hypothesized to have unified architectures for two different Use Cases: A and C.
Then, the overall architecture changed with respect to the one shown in figure 2.5, in order to differentiate the two scenarios considering that they have different actors (in terms of number but also in terms of actors themselves), different goals and therefore different architectures.

In the upper side of the architecture shown, there are many AGVs (most of them are from Omron), a Kuka cobot (also called rich AGV), a Franka cobot, a PLC (Programming Logic Controller) and System Managers.
From MES or WMS, that are company proprietary softwares which interact with each other, some work orders are received, respectively for the shop floor and for the warehouse.

The middleware layer is a sort of bidirectional channel that translates and standardizes the communications between the smart factory and the devices.
By means of a OPC UA(Open Platform Communications Unified Architecture) that is an extension of OPC Classic and it is a communication standars that provides platform and vendor independent secure communications, commands are interpreted by the various System Managers in order to reach the robotics.

Each AGV is controlled by its System Manager, with its own APIs which can be open or closed.
PLC, instead, is used to send commands to devices which do not have a System Manager and its purpose is to translate high level commands into binary ones that are understood by robots.

The cloud image simply means that every information is always stored on the Reply Cloud for further analysis or is got from it, by means of RESTful APIs.

However, it is important to point out that the robotic arm, together with the mobile system it is endowed, needs to have some fundamental characteristics to be able to work under the conditions just presented. In particular it has to:

- Implement precise navigation algorithms, in order to correctly position in the shop-floor.
- Be able to distinguish among items to be picked and to detect them, by means of sensors or 3D cameras together with a computer vision system.
- Coordinate in a precise way (both the robotic arm and the mobile part) to be

capable of collecting items from the shelf in a secure way and to store them in the final kit (or possibly an intermediate kit)

AGVs are robots widely used in factories or warehouses to move pallets and boxes with the assets from one part of the warehouse (or shop floor) to another.

Figure 2.6: Example of AGV [20]

2.9.1 Working Phases - Main scenario

The main scenario of Use Case C is about kitting of semi-assembled parts for the assembly line of an engine.
Kitting automotive logistic is often manually executed by a warehouse worker and it became necessary because production is becoming more and more customizable by clients.
For each configured car, the parts to be delivered to the assembly line need to be picked up among a great variety.
Kitting is then carried out in a big dedicated area, called supermarket, where it is possible to pick all the parts that need to be successively mounted.
For each car to be produced, a single order is instantiated with the requested variations.
The warehouse worker picks up the parts from the supermarket and sends them to the assembly line in kit.

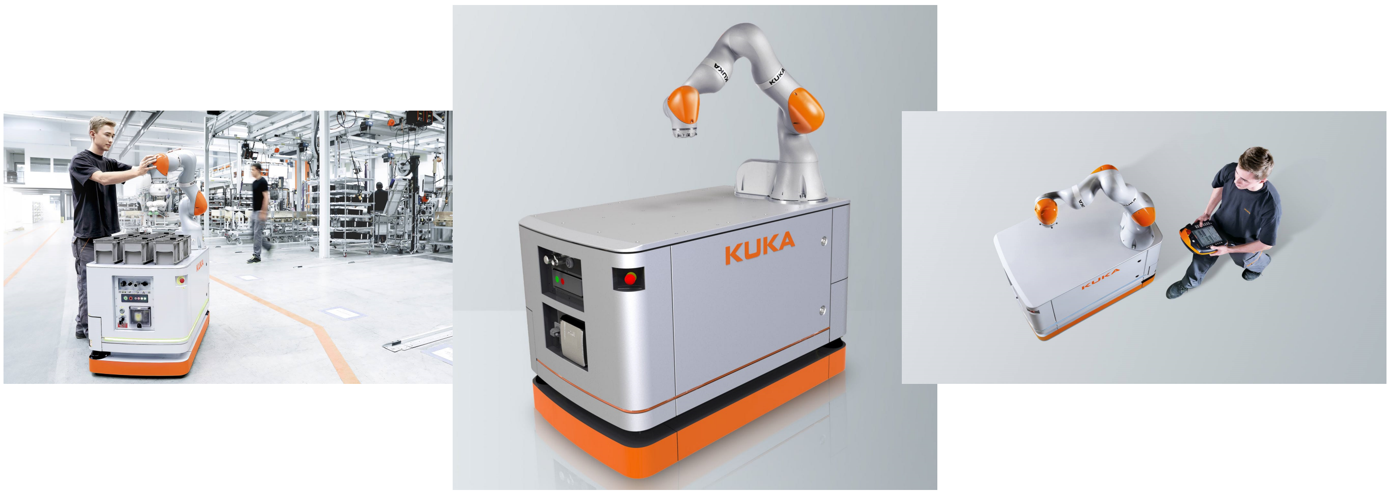


Figure 2.7: KUKA KMW iiwa [17], [23]

The scenario depicted by ICOSAF, whose aim is improving the already existent manual mechanism, is a collaborative work area where an operator and a cobot work carry out some tasks in an independent way, synchronizing when necessary:
The worker, thanks to a pick-to-light system, which is a system that allows shop floor operators to clearly understand in a quick way which item needs to be picked from a shelf lighting up the corresponding led, picks a number of items from a shelf in front of him (this number is defined in relation to the order, given by the MES, that is being fulfilled) following an orderly manner,as if each item were in a shopping list, and lays them down on a cart being carried.
In the meanwhile, a KUKA cobot, which can be briefly described as an AGV having a robotic arm on top of it, equipped with some computer vision logic enabling it to understand which object needs to be taken according to the shape and the positioning, picks other items from the shelf in front of it, following a given order and depositing them on itself.
This is possible thanks to its end effectors capable of moving and grabbing pieces.

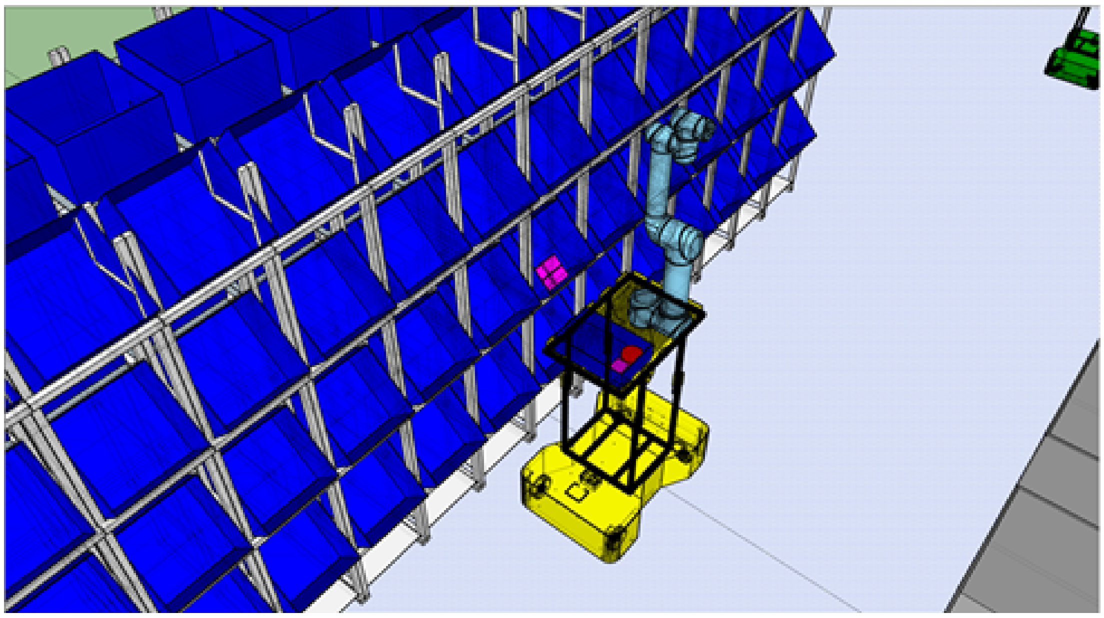


Figure 2.8: AGV KUKA Picking

At the end of this phase, KUKA positions itself over a stop point, waiting for the operator to finish the job.
Here the synchronization starts, in fact, KUKA must not move until it is released by some remote operator, monitoring this work area, or the operator on the shop floor frees it.

When the operator correctly collected all the required pieces, the next step will be to go to the kitting node in order to unload all the items picked by KUKA, putting them inside his own cart.

At the end of this process both KUKA and the worker can go back to their starting position in order to work on the next order scheduled.
An AGV eventually reaches the cart, seizes it and brings it where it is required to.

All the timings, intervals between one pick and another, positions of the actors, picking delays and so on, have been accurately defined by the project partners taking into account statistical data in order to generate meaningful and appropriate values.

The reasons this scenario improvement was thought can be found in the reduction of worker's NVAA (non value added activities), the enhancement of the workspace's ergonomic index, simplification of the picking process and the exoneration of the most repetitive and burdensome operations that the operator had to carry out, by means of the collaborative robot, so that the worker can concentrate on supervision and manipulation of the most delicate components.

The following journey map has been produced by the web designer of the project group, according to the CRF requirements.
At the beginning of the development phase, in fact, this map has been modified and discussed in order to properly understand whether it was possible to optimize and improve the suggested behavior or not.
The actors involved in it are, of course, the same involved in the main scenario: the control system (in gray) in charge of collecting all the data coming both from the simulation environment and from the website and it represents the logic behind the virtual copy and the web application, the operator in charge of kitting (in green) and the KUKA cobot (in blue).
The orange part represents the error scenario that will be discussed more in details in the next section.

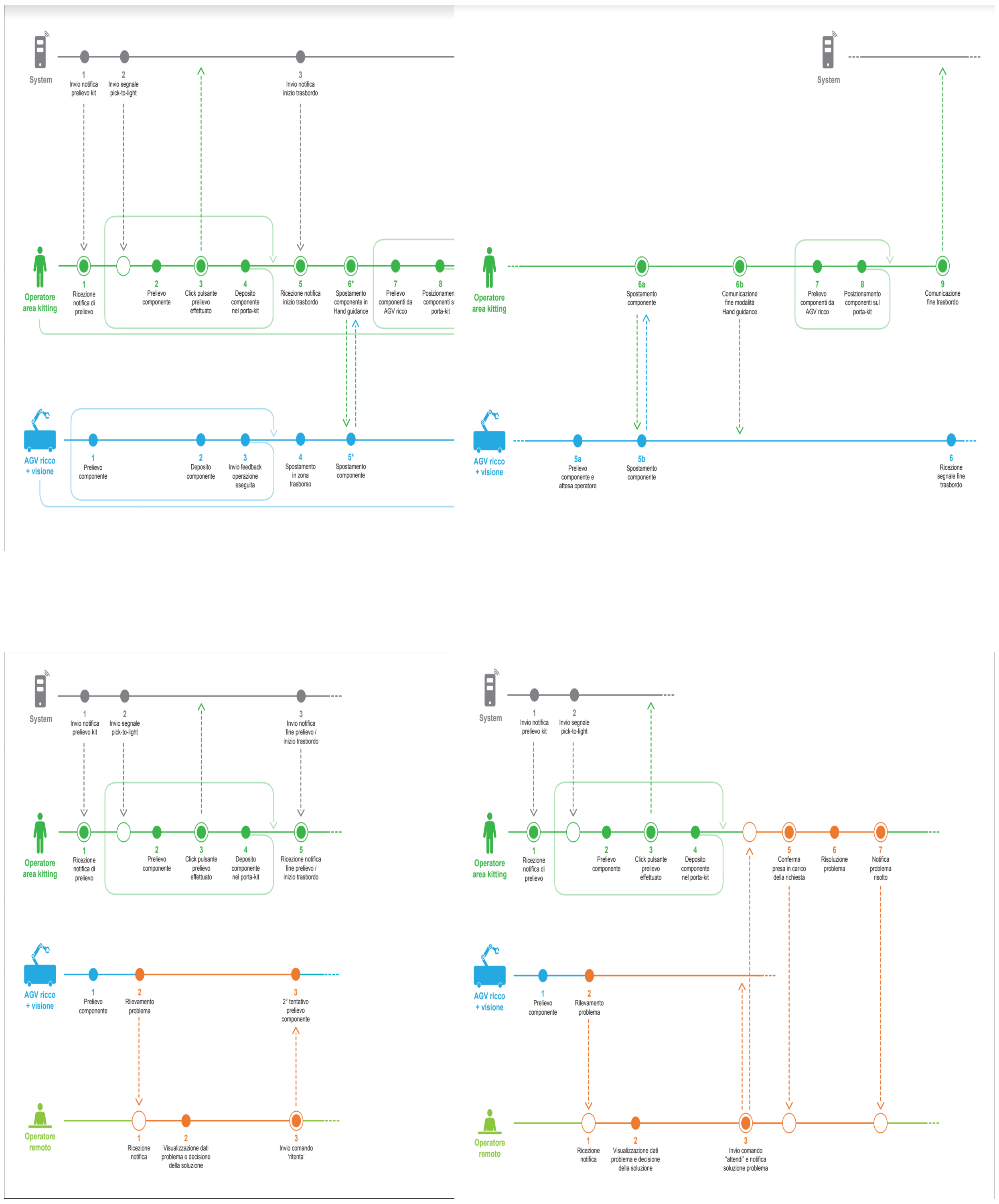

System
1 Invio notifica prelievo kit
2 Invio segnale pick-to-light
3 Invio notifica inizio trasbordo
Operatore area kitting
1 Ricezione notifica di prelievo
2 Prelievo componente
3 Click pulsante prelievo effettuato
4 Deposito componente nel porta-kit
5 Ricezione notifica inizio trasbordo
6* Spostamento componente in Hand guidance
7 Prelievo componenti da AGV ricco
8 Posizionamen componenti s porta-kit
AGV ricco + visione
1 Prelievo componente
2 Deposito componente
3 Invio feedback operazione eseguita
4 Spostamento in zona trasborso
5* Spostamento componente
System
Operatore area kitting
6a Spostamento componente
6b Comunicazione fine modalità Hand guidance
7 Prelievo componenti da AGV ricco
8 Posizionamento componenti sul porta-kit
9 Comunicazione fine trasbordo
AGV ricco + visione
5a Prelievo componente e attesa operatore
5b Spostamento componente
6 Ricezione segnale fine trasbordo
System
1 Invio notifica prelievo kit
2 Invio segnale pick-to-light
3 Invio notifica fine prelievo / inizio trasbordo
Operatore area kitting
1 Ricezione notifica di prelievo
2 Prelievo componente
3 Click pulsante prelievo effettuato
4 Deposito componente nel porta-kit
5 Ricezione notifica fine prelievo / inizio trasbordo
AGV ricco + visione
1 Prelievo componente
2 Rilevamento problema
3 2° tentativo prelievo componente
Operatore remoto
1 Ricezione notifica
2 Visualizzazione dati problema e decisione della soluzione
3 Invio comando 'ritenta'
System
1 Invio notifica prelievo kit
2 Invio segnale pick-to-light
Operatore area kitting
1 Ricezione notifica di prelievo
2 Prelievo componente
3 Click pulsante prelievo effettuato
4 Deposito componente nel porta-kit
5 Conferma presa in carico della richiesta
6 Risoluzione problema
7 Notifica problema risolto
AGV ricco + visione
1 Prelievo componente
2 Rilevamento problema
Operatore remoto
1 Ricezione notifica
2 Visualizzazione dati problema e decisione della soluzione
3 Invio comando "attendi" e notifica soluzione problema

Figure 2.10: Journey Map

2.9.2 Working Phases - Error scenario

What if during the picking phase, Kuka encounters some issue?
First of all, the collaborative robot should stop its execution and will stop moving in order to guarantee the security of all the actors involved, staying blocked until some action is performed by a remote operator, who is in charge of controlling all the work areas.

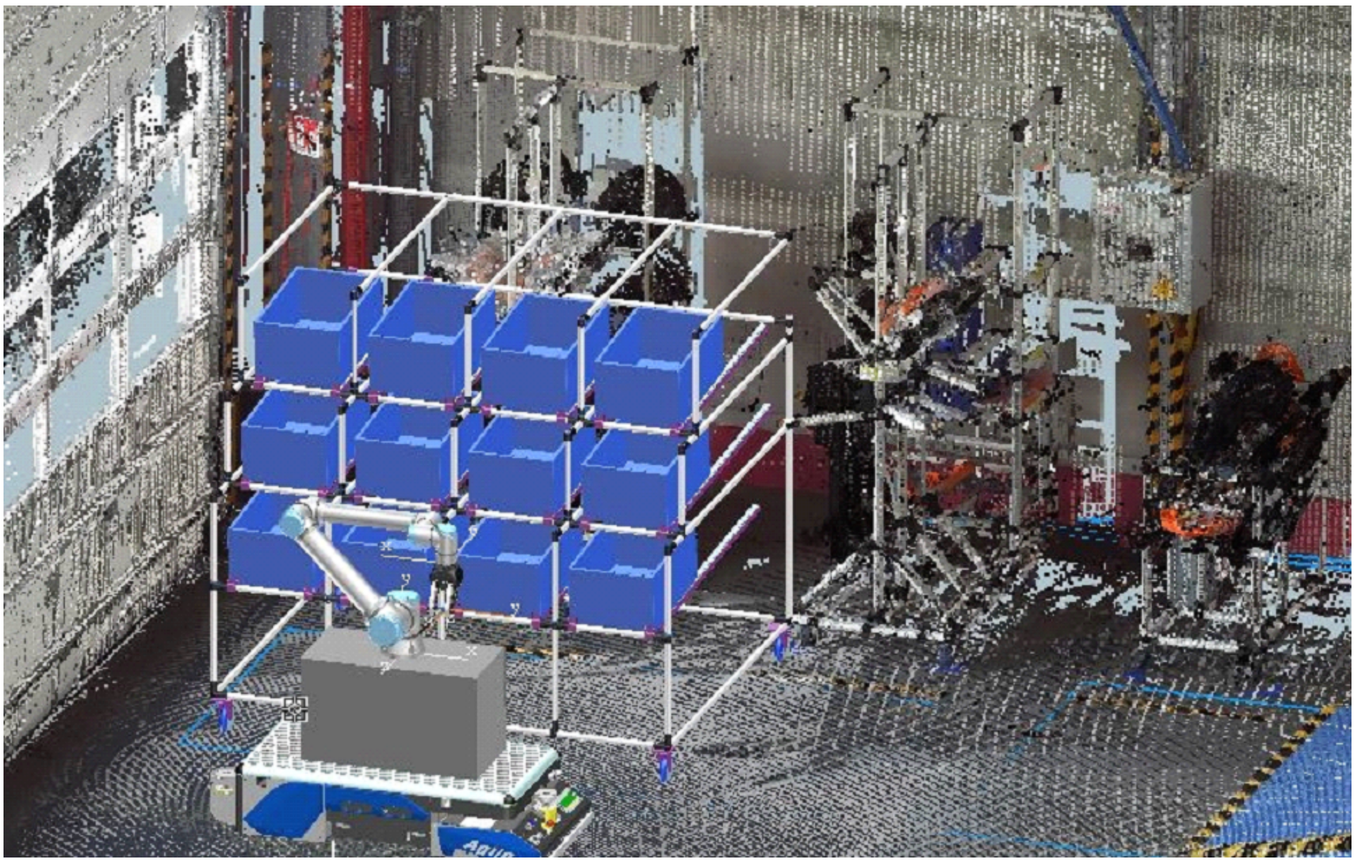


Figure 2.11: Kuka shelf

By means of a web user interface, the remote operator will solve the corresponding issue, by sending some commands to the on field operator or, directly, to the collaborative robot itself.
In this case, Kuka can retry the pick that led to the failure or simply wait for someone (the shop floor operator) to follow the steps indicated by the remote operator to solve the problem. Otherwise, it would be possible also to let the collaborative robot resume its job as if nothing happened before.

2.10 Collaborative Space

The system that must be developed is about a collaborative space in which a robot and a human operator need to collaborate to reach a common goal with the provided (by CRF) mode of operation.
Hence, it is important to define what is a collaborative space and how a human and a robot may work together.

Human-Robot integration in the context of a collaborative space exists when they both coexist actively within a space without physical barriers. Therefore, their integration must not be interpreted as a direct contact since the involved subjects can interact between each other also by means of spatial and/or temporal separation.
As for the robot, analyzing the desired interaction, then it emerges the need for selecting the strength, power, load and velocity compatibly with the operation that must be executed.Other factors that characterize the interaction between the two actors mentioned above, are the control system performances, robot moving limitations, stop mode, the path followed by the robot, environmental conditions in which it is working, the typology of end-effectors and workpieces.

The current regulatory framework identifies the possibility of carrying out human-robot collaboration on the basis of four operating modes:

1. Safety-rated monitored stop: robot protectively stops when its security perimeter is violated or when the operator enters the collaborative space and it stays in this state until the operator leaves.
2. Hand-Guiding: the operator exploits a device in order to move the robot's end effectors and it is used to execute manufacturing or to program the robot motion.
3. Speed and separation monitoring: robot and operator can work simultaneously in the same collaborative work space maintaining a minimal security distance.The robot adapts slowing down its movements whenever the relative distance decreases.
4. Power and force limiting: voluntary and accidental physical contacts between robot and operator are allowed.There is no minimal security distance to be respected, but the robot moves as slowly as possible.

As for Use Case C, KUKA and the operator will work in time and workspace-sharing mode, meaning that they will share at the same time the same collaborative area even though, to be more precise, they will work on different tasks but they will synchronize at some point.

Chapter 3

HMI

A HMI is a layer between the user and the machine, process, device or system being controlled. It is an interface which permits to interact with the objects listed before.
Even though HMI is the most common terminology, there are a few synonymous that are often referring to the same thing, such as Man-Machine Interface (MMI) or Operator Terminal (OT).
HMIs are widely used in companies as advanced interfaces to control machines or systems, but industries which make use of this kind of system, exploits them for different reasons. Human Machine Interfaces, in fact, can display meaningful data, track production processes, time and machines consumptions in terms of energy, monitor machines input and outputs and so on.
Yet it is important to clarify that HMI is not only GUI (Graphical User Interface). The main difference lays on the interfaces an HMI is provided. It is possible to say that a Human Machine Interface in most cases contains a GUI, but it is a complete control system.
The goal of ICOSAF, with respect to human machine interfaces, is to find a way of interacting with collaborative robotics and to assist workers in manual operations. Moreover it is necessary to develop new HMI systems able to interact effectively with AGVs for the logistics and to display logistics related data in real time.

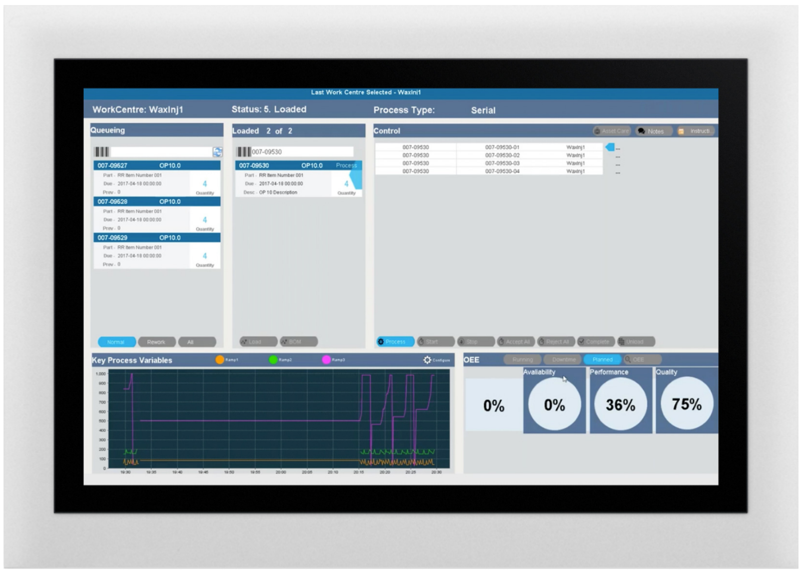


Figure 3.1: HMI Example[7]

3.1 Features of HMI

There are three main aspects that need to be considered when analyzing the state of the art of HMIs:

- User Experience and Usability
- Technology
- Cognitive Load and other factors

3.1.1 User eXperience and Usability

According to Garrett [16] the user experience design process of a software product is made up of five steps, that can be thought of as five different planes. These planes are stack one on top of the other and the one which is under the other represents its basis.

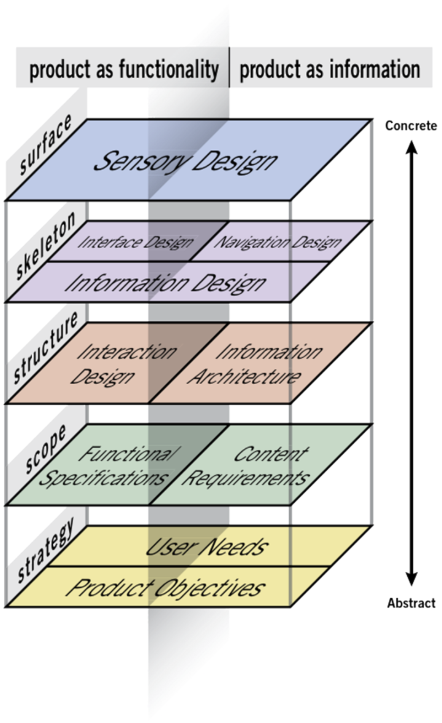


Figure 3.2: Garrett's Planes [19]

The strategy plane is the first that must be taken into account during this process and it is about both business objectives and user needs.
This phase embraces the target users discovery phase and the needfinding one. While for the first one a decision must be taken, considering the context where the users find themselves, the moments in which the product will be used and so on, the latter can be carried on after interviews, focus groups, observations and/or diary technique.
This plane is about the questions "why should we do it?", "which is the purpose?"

The scope plane includes what features the product should have. It is a sort of list of the functionalities that will be offered by the final product. It answers the question "What will the product do?"

The structure plane is about the logical structure of the product. For example, the navigation structure can be defined at this step.

Skeleton is the basic implementation of the product itself. It is in this moment that the position of the graphical elements is defined.

Surface includes whatever content the user will see (including colors, animations, effects and so on) and the kinds of interactions that will be present in the product. It is about sensory design.

Each of the planes described, could not be addressed if the previous plane had not been taken into account before. The decision taken in each plane, determine the choices that will be available in the further analysis on the next plane above.

What about the evaluation phase?
In order to test whether the product will be considered usable by users, a usability testing should be carried out. Furthermore this kind of test is useful to speed up this project phase and it produces cost savings. There are different ways of testing. It is possible to have user testing (usability testing or controlled experiments are examples of this strategy), where real users belonging to the target selected will actually use the system to give feedbacks and impressions about it. However, one of main techniques used is identifying and taking corrective actions against usability issues as soon as possible and it is based on expert evaluation.
Experts will evaluate the product referring to a set of heuristics (often used the Nielsen's Usability Heuristics [24]) to identify possible violations of them.

Heuristic number (#)	**Description**
1	Visibility of the system status
2	Match between system and the real world
3	User control and freedom
4	Consistency and standards
5	Error prevention
6	Recognition rather than recall
7	Flexibility and efficiency of use
8	Aesthetic and minimalist design
9	Help users recognize, diagnose, and recover from errors
10	Help and documentation

Table 3.1: Nielsen's Heuristics

3.1.2 Technology

According to the study conducted by J. Turesson [29],companies ask for some features they deem important for the Operator Terminal. Among them, IP-65 support is requested so that the terminal is dust and water resistant. There are also temperature-related features, like the support for dynamic changes or the high temperature resistance. Moreover, the presence of touch screen as well as color screen, clear visibility in sunlight and large screen size are considered very important.
From a more technical point of view, an easy and fast configuration, programmability, customizability and security are the most important features.

Function / feature	**Average importance (5 = important, 1 = less important)**
IP-65 support	3.83
Dynamic temperature support	3.17
UL-583 listing	2.67
High temperatures (>55°)	2.33
Low temperatures (<-10°)	1.83

Table 3.2: Ratings of contextual features for an Operator Terminal

Function / feature	**Average importance (5 = important, 1 = less important)**
Touch screen	4.67
Color screen	4.50
Clear visibility in sunlight	3.17
Large screen size (>= 10”)	2.33
Small screen size (<= 4”)	1.67

Table 3.3: Ratings of important features for a screen

Function / feature	**Average importance (5 = important, 1 = less important)**
Easy and fast configuration	4.83
Programmable	4.33
High level of customization	4.00
Several levels of passwords (N.B. only one respondent)	4.00
Compatibility with earlier models of the Operator Terminal	2.00

Table 3.4: Ratings of general important features for an Operator Terminal

3.1.3 Cognitive Load and other factors

Figure 3.3: Cognitive Load [28]

The cognitive load represents the mental effort needed to accomplish a certain task in terms of working memory usage. For a user interacting with a user interface, it should be reduced as much as possible. The cognitive load analysis was conducted by another project partner, starting from the assumption that the human behavior is generally regulated by two fundamental principles: the cyclic nature of human cognition and the dependency of cognitive processes on context and work environment. Industry 4.0 paradigm brings implications about the creation of company value, business models, services and work organization. As a consequence, the employees will have to work on modified processes and business models as well as interacting with the new technologies and collaborating with production resources. This implies the development of new learning models and competencies. In the scenario of collaborative areas, there are also other factors that contribute to the HMI design process, such as security. Starting from the assumption that protective barriers, separating human work areas and robots work areas, are useless in this case, this implies a big space saving and also increases the productivity, since collaborative robots can perfectly work when one or more operators is present in its work area. If on one hand human - robot integration offers to industries new alternatives that are more flexible and efficient with respect to the classic production lines, on the other hand introduces new problems related to the security aspect, as humans and robots will work in close contact and they need to understand one another, in order to avoid lack of security conditions, respecting system performances.

3.2 Brief analysis: choosing a device

On the basis of the aspects just considered, since HMIs come in a number of different forms and features, it is possible to conduct a brief analysis in order to distinguish among the existing technologies being able to choose the device that could possibly be the best one for the purpose of ICOSAF.
The main technologies that will be compared are:

- Smartphone / Tablet
- Smartwatch
- Wearable Glasses
- Smart Gloves

Smartphones interfaces are generally very clear and easily intelligible as nowadays they are widely used by almost everyone.
The presence of the touch screen (and the consequent presence of different gestures) and the color screen are taken for granted.
Screen size must be chosen in order to set a balanced trade-off between easy handling (feature that often presupposes the usage of a small screen sizes) and the ease of access to the needed information (which presupposes, on the contrary, bigger screen sizes).
The cognitive load associated to a mobile interface which exploits a smartphone is sustainable, as smartphones are commonly used nowadays by almost everyone more than the other devices under analysis and hence they rank as the most suitable with respect to the others from this point of view.

Considering a more technical point of view, smartphones perform well in all the aspects considered above.

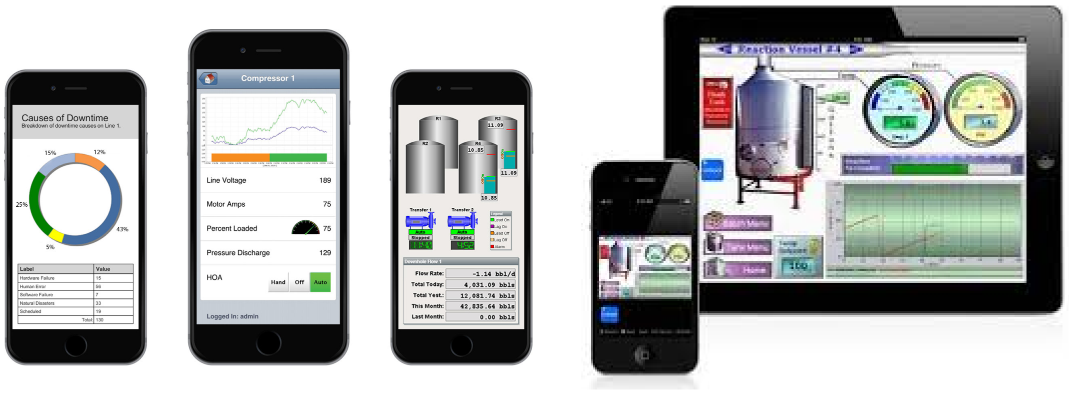


Figure 3.4: Smartphone / Tablet HMI [1], [15]

As for smartwatch interfaces, their size result to be less cumbersome for the operator to deal with, but still having almost all the most important characteristics pointed out previously for human-machine interfaces such as the presence of the touch screen and the color screen.
Nevertheless screen size is limited by the dimension that a wearable device must have, which in turn limits the number of gestures available.
Moreover, the cognitive load could be higher with respect to the "competitors", due to the large number of operations needed in order to accomplish a task or a page in particular.
From the technical point of view it is very similar to a smartphone, except for the security aspect, in fact, writing a password (or more than one) could result to be inconvenient.

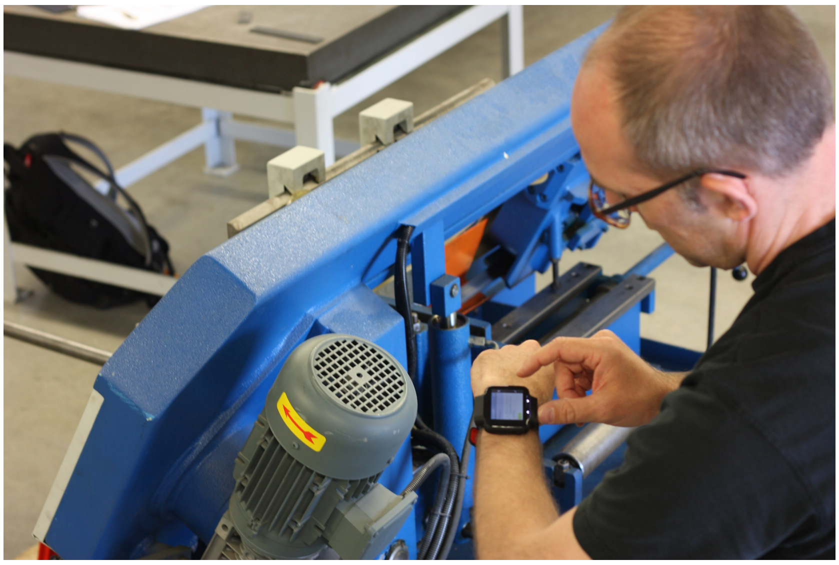


Figure 3.5: Smartwatch HMI [22]

Smart glasses are probably the most flexible choice, yet limiting for the operator because of the total immersion in a virtual environment (augmented reality). Working memory is sustainable because interfaces are very often customizable. From a more technical point of view, it would be impossible to insert a password without the implementation of a virtual keyboard, which would result uncomfortable in most cases.

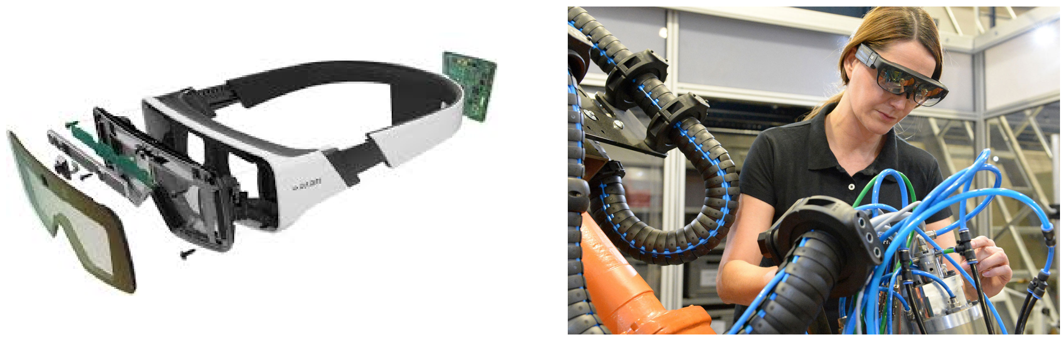
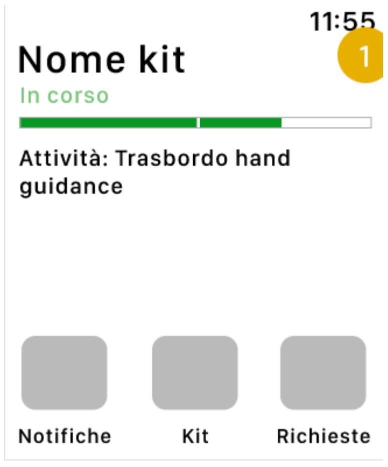


Figure 3.6: Smart Glasses HMI [2], [21]

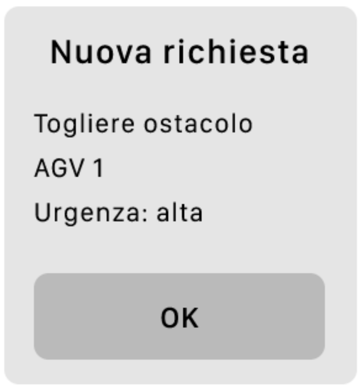
For what concerns smart gloves, they have been taken into consideration only in the first steps of the project by other project partners, but they have been discarded as they clearly do not comply with the requirements.

3.3 Final Choice

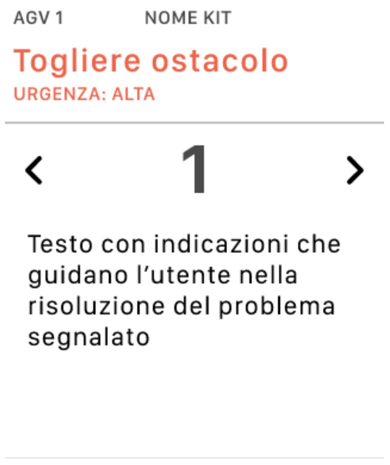
The final choice taken was to have operators who work in the shop floor to use a smartwatch since it is much more comfortable to be used on the field, having to deal with machines, robots and the required stuff related to their job.
Some wireframes of the possible implementation of this HMI can be found in figure 3.7. On the other hand it was chosen to exploit a bigger size screen for the remote operator who needs to have the whole situation under control with diagrams and schematics, too.
For these reasons the PC/Tablet/Smartphone option was the one that best fitted the requirements and this will be the HMI this thesis work will focus on.



(a) Homepage



(b) Request Notification


AGV 1
NOME KIT
Togliere ostacolo
URGENZA: ALTA
<
1
>
Testo con indicazioni che guidano l'utente nella risoluzione del problema segnalato

(c) Steps to solve the issue

Figure 3.7: Smartwatch HMI Wireframe

3.4 Dashboard UI integration goal

Dashboard design is a simple view that presents all information, shows trends and risky areas, updates users on what happened and preview of the most crucial information for the user at the moment he is looking at it.

Different types of dashboard definition exist in the modern UX design world,but the main ones are:

- Operational dashboard aims to bring critical information quickly to users.
  Its main goals is to present data deviations to the user as fast as possible and clearly, showing the needed resources as well as displaying their status.

- Analytical dashboards, instead, provides the user with at-a-glance information used for analysis and decision making.
  One of the main objectives of this kind of dashboard, as it is mainly thought for decision making, is to help users make the best sense of data, analyze trends and drive their choices.

IoT connectivity provides the manufacturing sector with an increasing number of ways to access sensor-driven data from industrial machines and equipment and/or from monitoring goods movements inside the shop floor.
Being able to acquire the data from the factory floor and to perform some level of analysis is an important part for any IoT system.
The data collection is typically focused on aggregating raw data from different endpoints, then filtering and/or processing the raw data so that it is available for real-time or batch analysis.
The system manages the sensor measurements and performs light data filtering and aggregation of data. Then it collects data from multiple devices and distributes it in accordance with settings.

The technique used to develop this UI was to build a Single Page Web Application (SPA), exploiting the Angular framework.
It was chosen to implement a SPA in order to increase the performances and not to reload the page each time a link or a button is clicked.
While data is being exchanged, the DOM structure and its style is set only once and this makes everything faster with respect to a traditional web application.

Web services can be used by means of servers that export data through Web APIs, in fact, RESTful web APIs can be accessed via standard HTTP methods by client browsers and mobile devices, such as GET or POST.
As for the UC-C, a SQL database was exploited (with a MySQL DBMS).
Services are written in nodeJS and they make RESTful APIs available to the external.

The remote operator is going to visualize all the information needed, by means of a dashboard that will be realized as a web application.
Furthermore, some notifications will pop up in case of error, or in case some actions need to be performed.

The details about how the web applications should appear come from the web designer of ICOSAF, who is also in charge of realizing wireframes, sketches and high-fidelity prototypes.

This dashboard will be exploited by the remote operator who needs to have the control over every work area, in fact, it is possible to see the situation of each AGV at a glance and some charts explain what is happening more in details in order to have more precise statistics that could lead to some deeper insights.

Figures 3.8 and 3.9 show how the web application was designed to be by the web designer. Starting from this prototype, the actual application was developed.

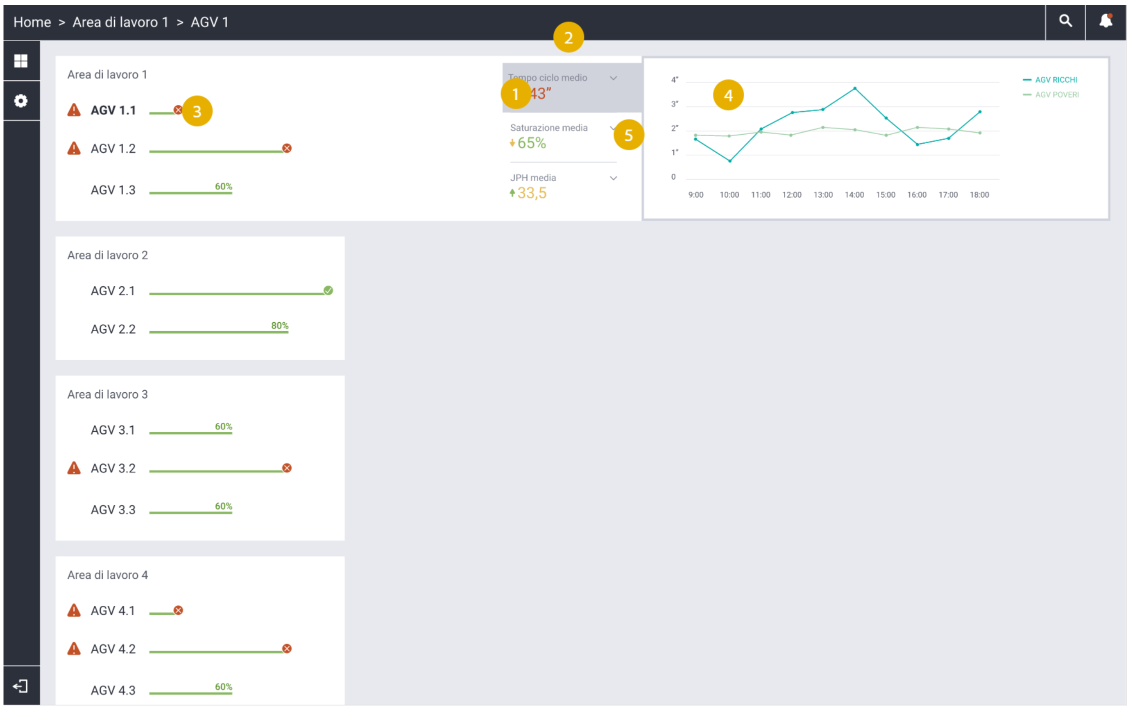
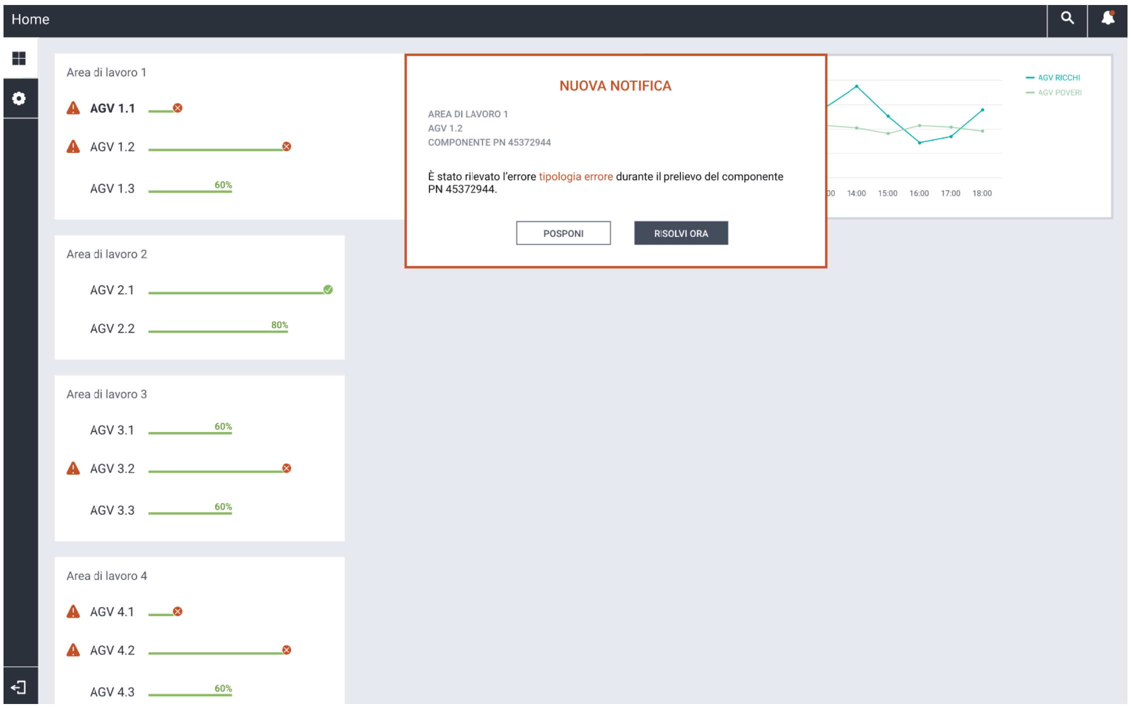


Figure 3.8: Web Application prototype



NUOVA NOTIFICA

AREA DI LAVORO 1
AGV 1.2
COMPONENTE PN 45372944

È stato rilevato l'errore tipologia errore durante il prelievo del componente PN 45372944.

POSPONI RISOLVI ORA

Figure 3.9: Notification example

3.5 State of the art - Web Development Frameworks

Let us dive deep into the state of the art for the development of a web application to implement the desired features.
The focus will be the implementation of the dashboard UI, since the smart watch interface still needs to be implemented and properly described to have a clear idea of what its aim should be, even though it was already chosen to exploit that kind of device for operators on field.
For the dashboard UI, instead, not only the device to be used has been defined (Tablet/PC), but also its aspect in terms of layout (thanks to the wireframe) and the main information that need to be displayed such as the overall situations of the work areas, the details for an AGV, the handling error part and much more have already been chosen.

Starting from the assumption that web pages development has undergone a huge change during the years, the idea of using PHP to code dynamic web pages written in just HTML and CSS was immediately discarded in favor of a framework based on JavaScript that avoids coders to self write boilerplate code and moreover exploits software design patterns as MVC, which is much more intuitive with respect to the other solutions and so on.

The framework that was chosen is Angular.

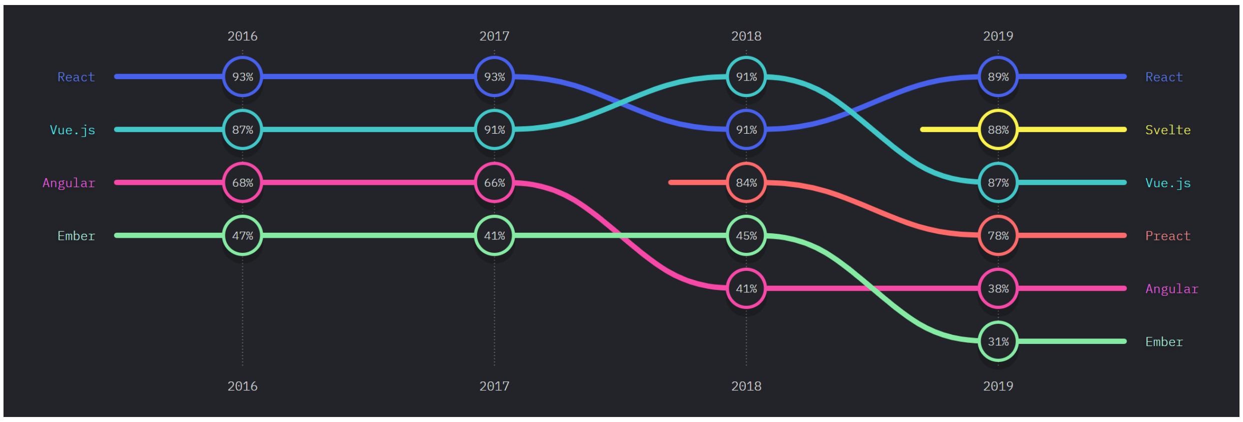


Figure 3.10: Front End frameworks - Satisfaction [14]

Even though figure 3.10 points out that the satisfaction rate is much higher for users developing with React, other factors need to be considered such as the awareness of a framework with respect to another and the interest demonstrated by coders in it.

Considering the project stage in which this thesis work is located and the fact that all these frameworks could perfectly fit the ICOSAF requirements, it was chosen to exploit a technology which was already known and Angular was the one whose knowledge was more spread among the project developers and partners.
Angular is, in fact, widely used in big companies and it is an open-source framework developed by Google for single-page applications.

3.6 Angular

Figure 3.11: Angular [30]

Angular is a framework, as explained before, that was created to let programmers develop in a easy and fast way applications that can be deployed on a variety of platforms, such as smartphones, tablets and PCs.
Differently from AngularJS, Angular is based on TypeScript, which is a front-end open source programming language from Microsoft which extends the JavaScript syntax that can be interpreted by any web browser or app.

Angular is based on components and each of them is written in three different languages, that are:

- HTML for displaying the elements in the page, enriched with Angular specific markups
- CSS for styling them, adding colors, setting positions and sizes
- TypeScript to handle the logic behind them, meaning the state components need to maintain, the way routing between them is computed, the data needed to populate the HTML components and so on.

A single page can be made of more components whose logic, HTML and CSS code is kept separated.

Another fundamental element in developing an Angular application is the use of services.
Services, in fact, are able to provide all the necessary data coming from the back-end, handling the communication with a DB.
Indeed, it is possible for a component to subscribe to a service in order to get possible variations of data and react consequently with an asynchronous callback which is called as soon as some changes happened.

Part II

Second Part

Chapter 4

Simulation

Simulating means virtually reproducing a physical system that already exists or that will be implemented in the near future. The aim is to define a model that perfectly fits all the requisites that will be needed by the real one.
So interfaces, communication mechanisms, hardware simulated need to be defined in such a way that the simulation perfectly reproduces its real twin.
The reason why simulating is so important is that it would be possible to optimize the performances of the real system or parts of it (like AGVs and cobots consumptions or time in which a value added task is performed), but also to prevent possible errors or issues, to understand the feasibility of the whole process (due to the fact that the simulation will receive some inputs from the same physical systems that will be interconnected to the real system once it will be realized, such as MES and WMS or simply receive similar inputs as if they came from those systems), and so on. Hence, simulating a system is certainly a concrete advantage in terms of cost savings, optimization, feasibility analysis and forecasting.
In order to virtually simulate the physical process described in 2.9.1, some steps have been followed. The simulation has been carried out with the AnyLogic tool and can be deconstructed in 5 main parts:

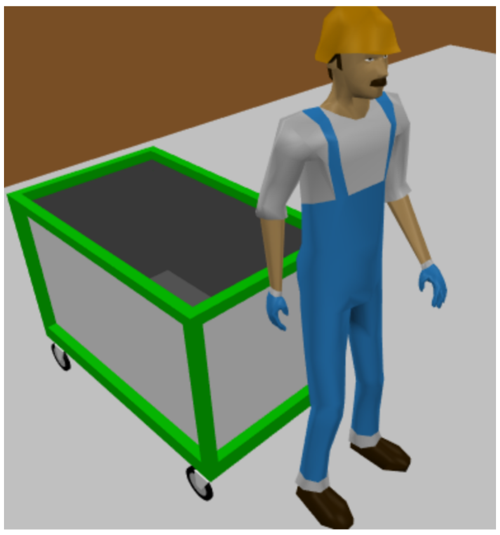
- Variables, Parameters and Actors definition
- 2D
- 3D
- Logic
- Statistics

Each of them will be discussed and explained in details.

4.1 Variables, Parameters and Actors

In order to build a simulation using AnyLogic the first requirement is knowing who are the actors involved in it, in order to virtualize them, giving them a 3D shape and a logic to be accomplished.
In the simulation of the Use Case C the actors involved are:

- Operator
- Kuka Cobot
- AGV


(a) Operator 3D

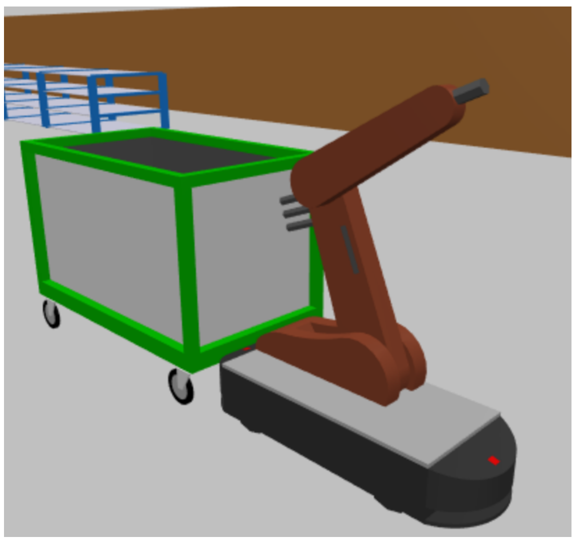

(b) Kuka 3D

Figure 4.1: Main Actors

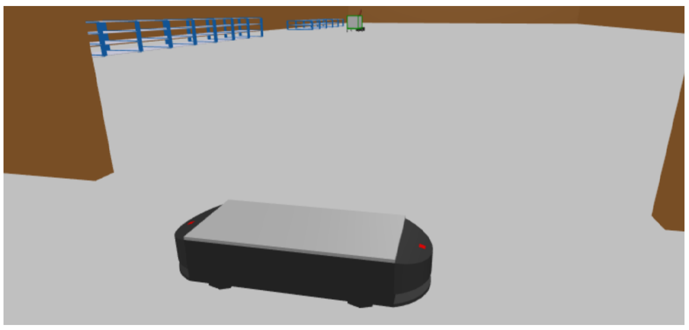


Figure 4.2: AGV 3D

4.1.1 Actors

In AnyLogic each entity must be defined, so in the list of actors also the industrial container (the element which will be picked from the shelves) and the carts (the objects in which each industrial container will be positioned at the end of each pick) need to be defined.
Creating an actor is quite simple and it is possible to choose some basic 3D representation of common objects or combine them, and then it is possible to set the initial position, the color, the orientation, the scale and much more.
It is needed to add a new Agent Type, choose a 3D shape picking it from existing libraries like People, Buildings, Rail Transport, Manufacturing and so on. Some parameters may be added at this point too.
When the actor has been created, another 3D shape can be added to the previously selected one.
As for Kuka, for example, three representations have been put together: a cart, an AGV and an arm robot, simulating its real shape.
Regarding the operator, instead, only the worker shape has been chosen and so also for the AGV.

4.1.2 Variables, Parameters and Functions

Throughout the definition of the logic some variables and parameters were used both as simulation configuration, meaning before the actual execution and as Java variables, hence during the execution.
There is actually one important difference between variables and parameters in AnyLogic, because the latter lets programmers define an asynchronous callback that will be executed as soon as the value stored in it changes.
The variables used as simulation configurations are respectively the delay of kuka when picking items and the delay of the worker. This is due to the fact that once those values are set, they can not be changed during the simulation, similarly to a real system, in fact, the picking time for the two actors involved is approximately constant over time.
Moreover, as shown in figure 4.3, some functions have been defined to navigate among 3D visualization of the simulation, 2D, logic and so on.
The other function is used to reset all the statistics when an order has been completely fulfilled and another one is being started.
Furthermore, some collections (like industrialContainersListKuka and palletRackListKuka) have been created to keep track of the list of industrial containers that Kuka and the worker need to pick (and in which order) and the list of pallet racks from which each item should be picked.

A little work around was introduced here because the packets, as requested by the project partners, had to be picked sequentially, meaning that each actor had to pick all the items from the nearest shelf, then all the items from the second nearest one and so on.
To implement this, 10 pallet racks have been inserted rather than one single rack system of equivalent available space, in order to keep track of the shelves that needed to be emptied first.

What about the variable clientRestAPI under the Kuka section?
As AnyLogic allows creating new Java classes, a clientRestAPI class has been created.
This class is in charge of contacting some APIs, getting the response, elaborating it and returning the expected values to the main.
Some methods of this class are used to ask for the order numbers to be fulfilled, ask for the list of tasks to be accomplished for each order or simply performing a GET to signal whether a pick has been correctly performed or not.
The most important method is listenService that will be explained more in details in the next chapters.

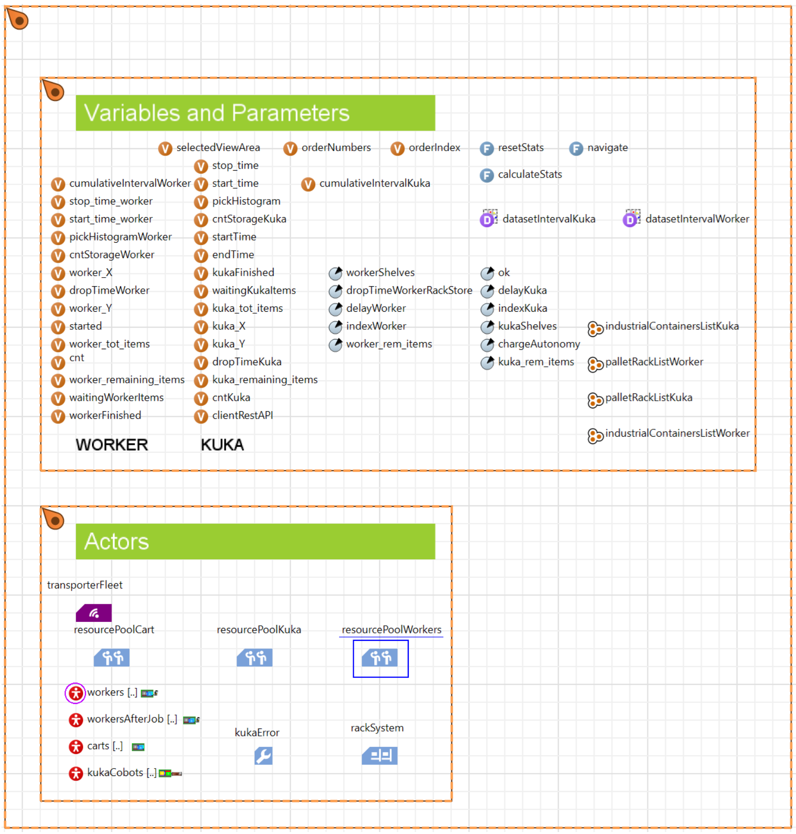


Figure 4.3: Variables, Parameters and Actors

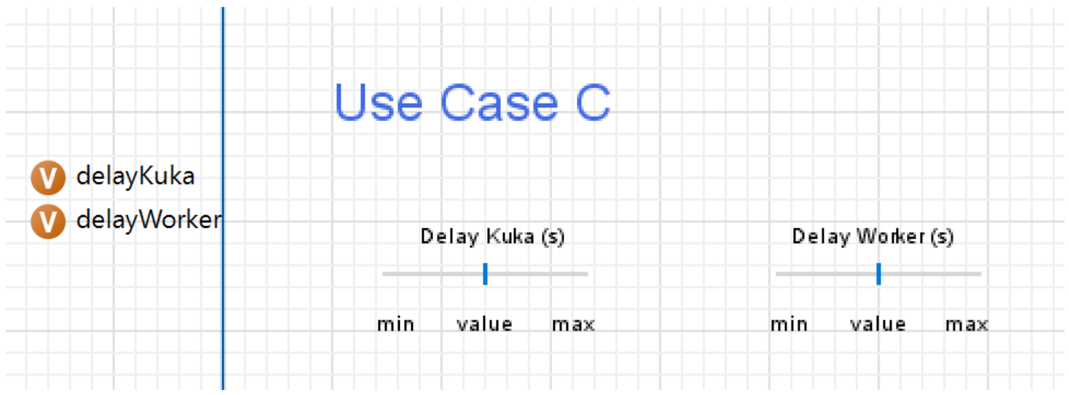
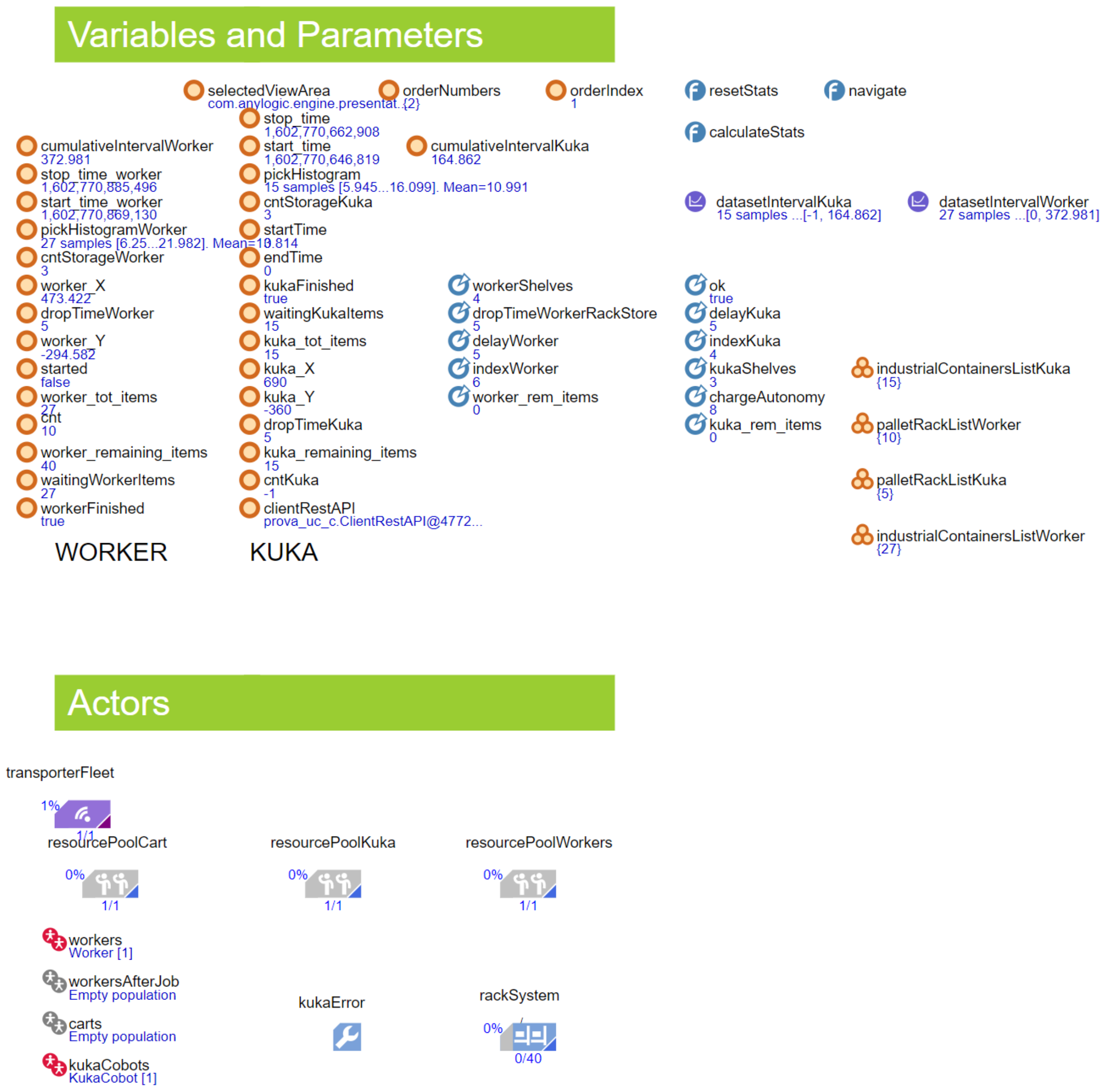


Figure 4.4: Simulation Variables

Figure 4.5 shows how the variable, parameters and actors section changed after the simulation.
Under each of them, in fact, some important value is written to let developers understand what is going on.
For example the dataset variables, that have been added to keep track of the picking time samples got both from the worker and Kuka, at the end of the simulation contain respectively 27 and 15 samples that perfectly match the tasks completed by them.
Variable "ok" keeps track of the status of the overall simulation and it is set to "true" because no error was present at the end of the simulation.
The variable that corresponds to the histogram, already contains all the samples and the mean has been already calculated and shown in this section. These and the other informations shown after the simulation could be very useful both for the testing and the developing phases.


Variables and Parameters

selectedViewArea
com.anylogic.engine.presentat.{2}
orderNumbers
orderIndex
1
resetStats
navigate
calculateStats

cumulativeIntervalWorker
372.981
stop_time_worker
1,602,770,885,496
start_time_worker
1,602,770,869,130
pickHistogramWorker
27 samples [6.25...21.982]. Mean=10.814
cntStorageWorker
3
worker_X
473.422
dropTimeWorker
5
worker_Y
-294.582
started
false
worker_tot_items
27
cnt
10
worker_remaining_items
40
waitingWorkerItems
27
workerFinished
true
WORKER

stop_time
1,602,770,662,908
start_time
1,602,770,646,819
pickHistogram
15 samples [5.945...16.099]. Mean=10.991
cntStorageKuka
3
startTime
endTime
0
kukaFinished
true
waitingKukaItems
15
kuka_tot_items
15
kuka_X
690
kuka_Y
-360
dropTimeKuka
5
kuka_remaining_items
15
cntKuka
-1
clientRestAPI
prova_uc_c.ClientRestAPI@4772...
KUKA

cumulativeIntervalKuka
164.862

workerShelves
4
dropTimeWorkerRackStore
5
delayWorker
5
indexWorker
6
worker_rem_items
0

datasetIntervalKuka
15 samples ...[-1, 164.862]
datasetIntervalWorker
27 samples ...[0, 372.981]

ok
true
delayKuka
5
indexKuka
4
kukaShelves
3
chargeAutonomy
8
kuka_rem_items
0

industrialContainersListKuka
{15}
palletRackListWorker
{10}
palletRackListKuka
{5}
industrialContainersListWorker
{27}

Actors

transporterFleet
1%
1/1
resourcePoolCart
0%
1/1
resourcePoolKuka
0%
1/1
resourcePoolWorkers
0%
1/1

workers
Worker [1]
workersAfterJob
Empty population
carts
Empty population
kukaCobots
KukaCobot [1]

kukaError
rackSystem
0%
0/40

Figure 4.5: Variables, Parameters and Actors after simulation

4.2 2D Representation

2D representation must be defined according to the warehouse under analysis.
In this case it represents two shelves one next to the other from which the worker and the collaborative robot will collect the industrial containers needed.
As for the operator shelf, it is composed of 10 pallet racks with 4 levels and 1 deep position for a maximum number of items equal to 40. For what concerns the kuka shelf, instead, it is made up of 5 cells with 3 levels and 1 deep position each for a maximum number of items equal to 15.

Moreover the stop points, the eventual conveyors and paths need to be defined at this level. In the simulation considered there is a wall surrounding the room, with a space which represents a passage, exploited by the AGV at the end of the simulation to pick up the final kit, with a predefined path.
Another very important issue that has been managed at this level is the camera. Its position in fact needs to be set in order to see the 3D simulation from a good starting point, without the need of navigating in it to find the perfect spot.

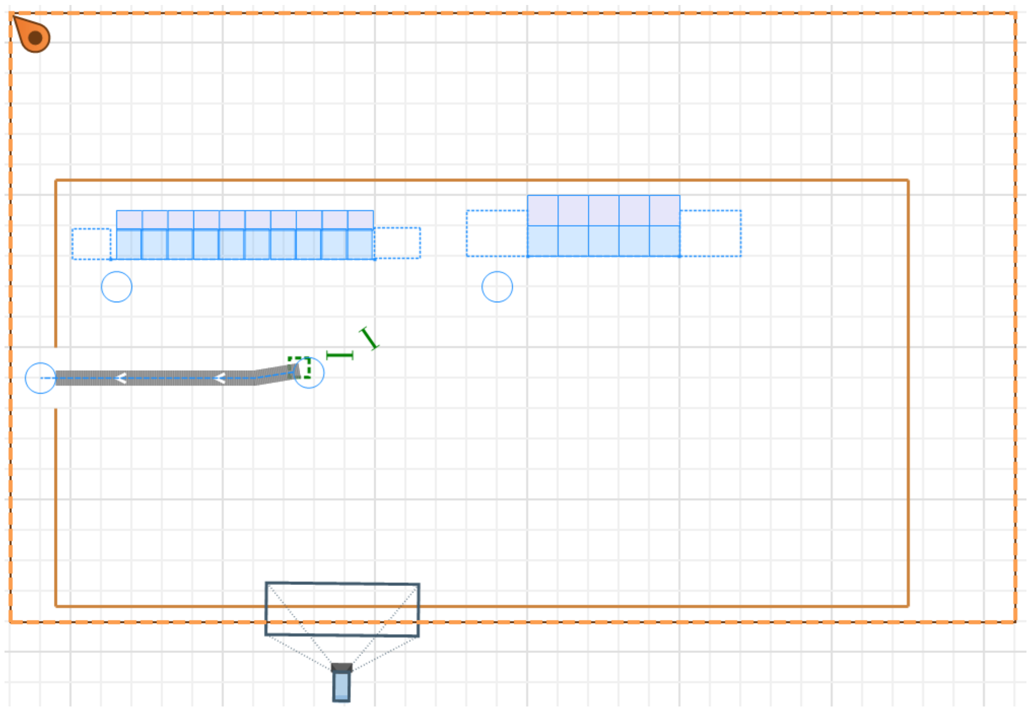
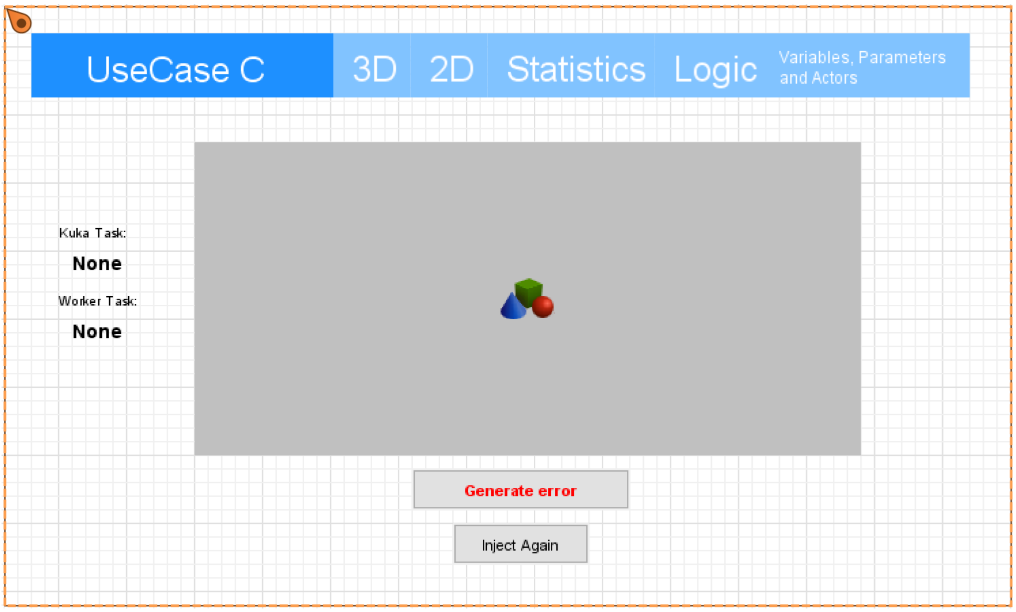


Figure 4.6: 2D

4.3 3D Representation

The 3D representation is automatically defined by AnyLogic starting from the 2D's one. What has been introduced is the possibility to easily switch between the different screens (2D, 3D, Logic, Variables, Parameters and Operators, and Statistics) by means of a simple click on a tab. The other two buttons visible report respectively the writings "Generate error" and "Inject again".
The first one is used in case it would be required to simulate an error happening on Kuka.
In this case the cobot immediately stops its execution and waits for an action to be executed by the remote operator. The remote worker, interacting with the UI dashboard, takes a decision (such as making Kuka go on in case it had an error related to a specific pick which went wrong or clicking a button to send someone to fix the issue).
In any case Kuka resumes its execution again when the action has been performed, contacting an API to get this kind of information.
The "Inject again" button, instead, is used to proceed with the next order scheduled for the current day, resetting all the previous statistics, variables, parameters and actors positions.
As it is possible to notice, next to the 3D block, there is an information (empty at the beginning of the simulation, when it has not been launched yet) about the task that is being carried out by the corresponding actor.



Kuka Task:
None
Worker Task:
None

Figure 4.7: 3D

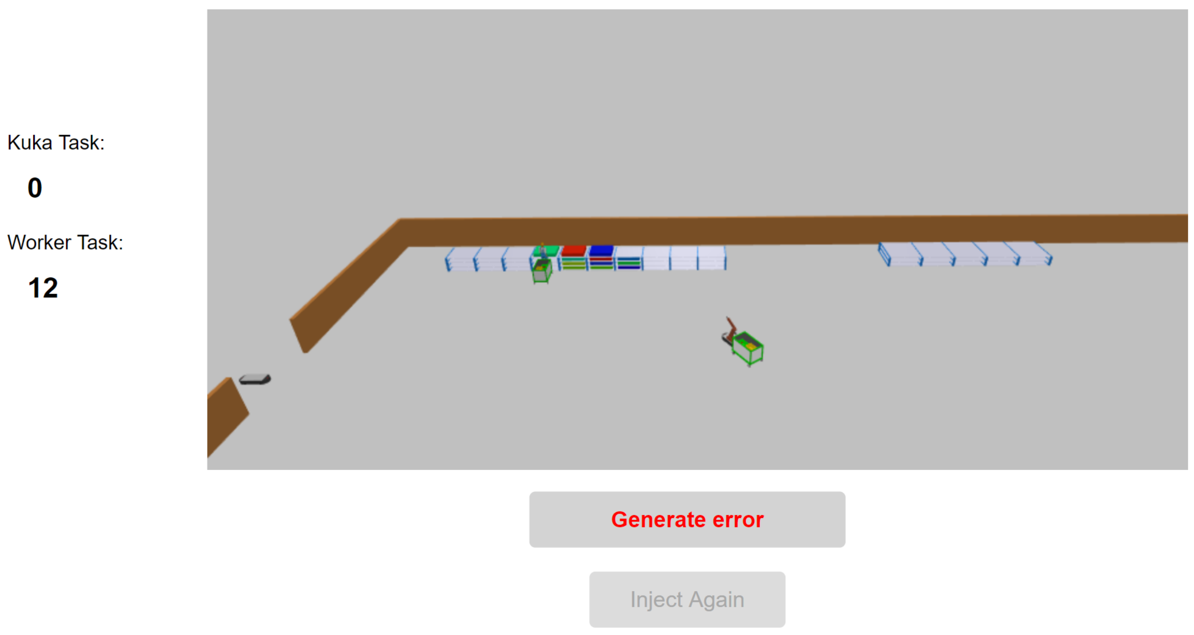


Figure 4.8: 3D after simulation

Figure 4.8 shows Kuka which already finished its picking phase and it is waiting for the operator to pick the item assigned to him.

Two numbers appeared on the left, representing respectively the task which is being carried out by Kuka and by the worker.
It was chosen to represent them in a descending order, to keep track of the number of task remaining: Kuka has no more tasks to complete (0), while the worker still needs to pick 12 packets.
Furthermore, the inject again button is not clickable because this order has not been completed yet.

4.4 Logic definition

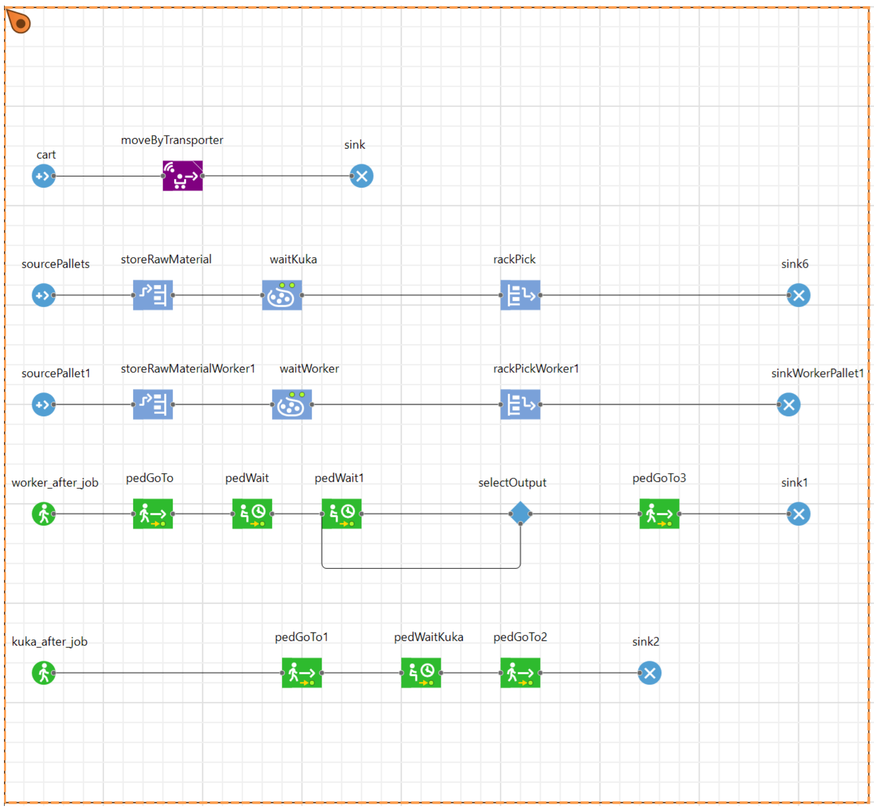


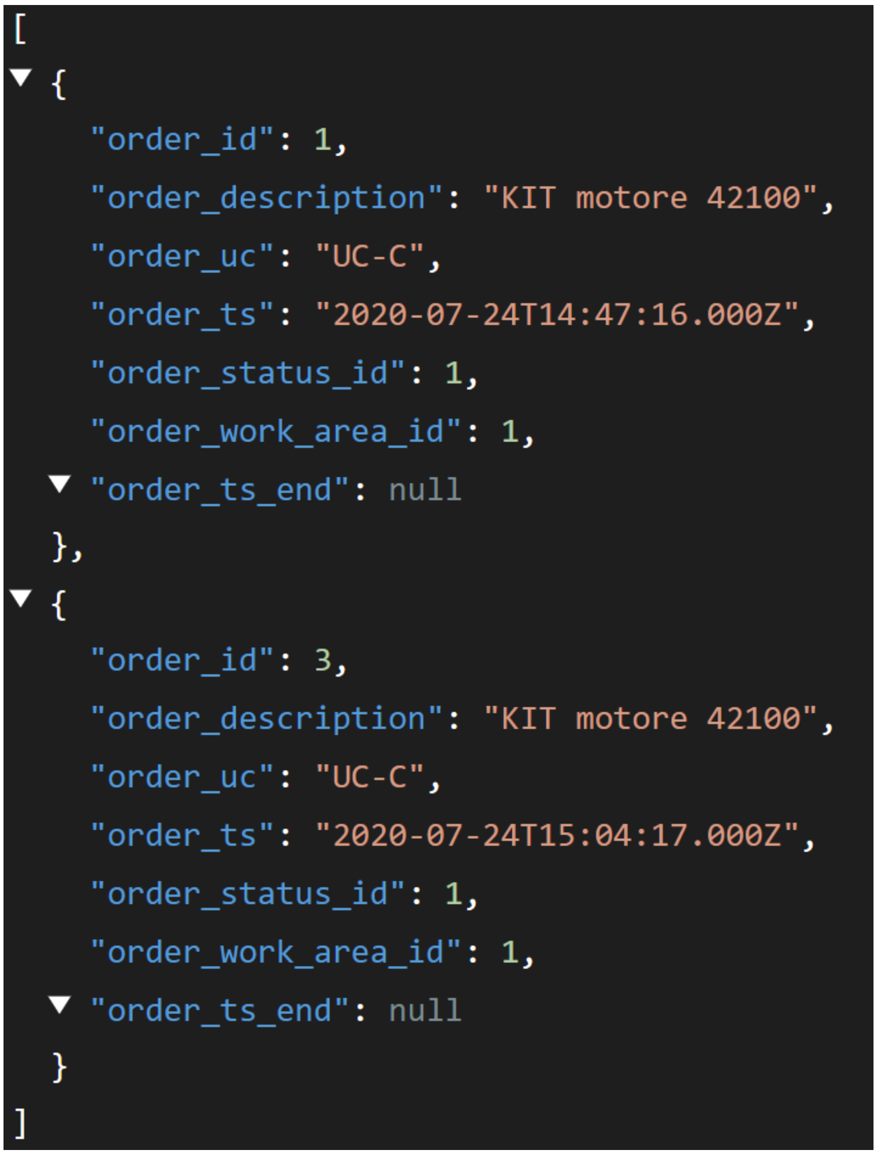
Figure 4.9: Logic

What about the logic?
This phase is undoubtedly the most important one. The interaction between actors, their changes during the execution, the usage of variables, parameters, functions and classes defined previously, the synchronization of actions, the measurements that need to be taken in order to develop some graphic explaining in more details what is going on, API calls and much more take place in this precise moment of the developing.

It is possible to exploit some predefined blocks with basic functions (such as the ones for picking items or for moving actors to specific nodes), but in order to have a complete and working solution some Java code is needed, enhancing those elementary blocks, to integrate most of the logic which is not already present.

At the main startup an API is exploited to get the orders that need to be completed for the selected date.

For this reason, the clientRestAPI class is exploited and within it, a method used for sending GET requests to custom URLs is called. A connection is opened and the input stream is processed.
The API called is the following:
http://icowms.cloud.reply.eu/Details/getOrdListbyDate?ts=2020-07-24&uc=UC-C
This API is called to get the orders that need to be fulfilled.
In this precise case, the outcome of a GET towards this URL would result in a JSON Array listing all the orders for the 2020-07-24.



```
[
▼ {
    "order_id": 1,
    "order_description": "KIT motore 42100",
    "order_uc": "UC-C",
    "order_ts": "2020-07-24T14:47:16.000Z",
    "order_status_id": 1,
    "order_work_area_id": 1,
  ▼ "order_ts_end": null
  },
▼ {
    "order_id": 3,
    "order_description": "KIT motore 42100",
    "order_uc": "UC-C",
    "order_ts": "2020-07-24T15:04:17.000Z",
    "order_status_id": 1,
    "order_work_area_id": 1,
  ▼ "order_ts_end": null
  }
]
```

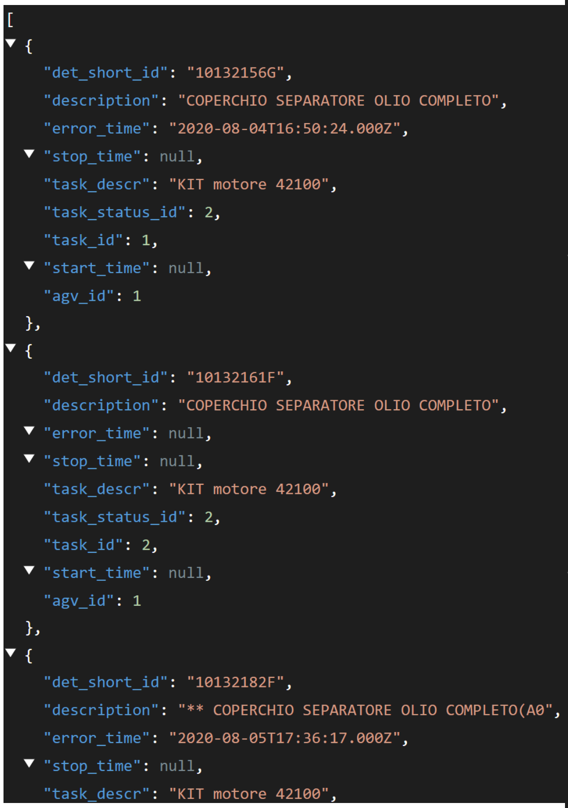
Figure 4.10: Orders

Once the orders have been collected, starting from the first one, by means of the call to the following APIs it is possible to get the tasks (that correspond to the items that need to be collected) respectively by the Kuka cobot and by the operator, specifying the orderID which is the ID of the order in execution:

`http://icowms.cloud.reply.eu/Details/getTaskListAGV?order_id="+orderId+"&agv_id=1`

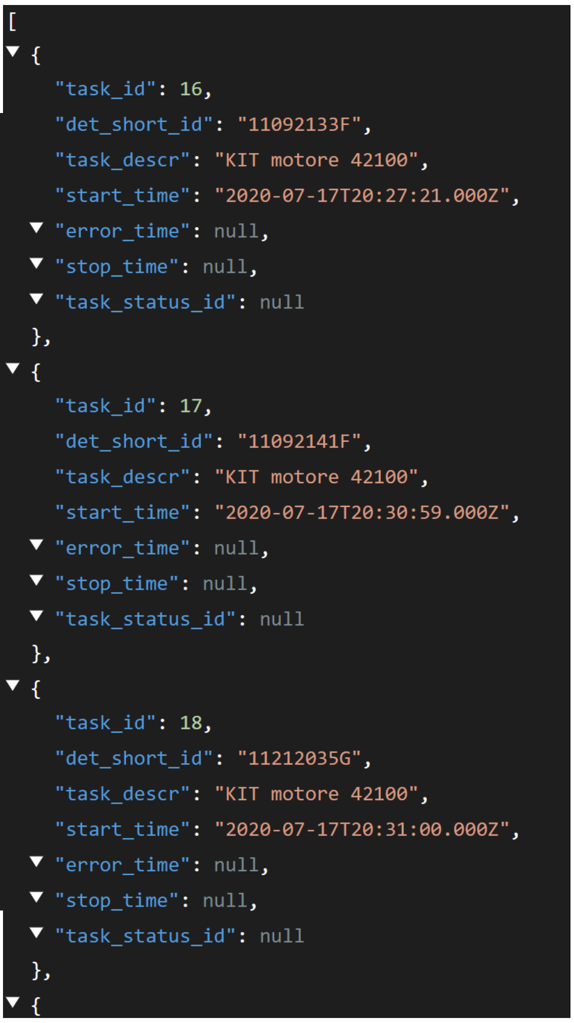
`http://icowms.cloud.reply.eu/Details/getTaskListOper?order_id="+orderId+"&oper_id=1`

This APIs are called to get the task list to be accomplished for a specific order ID respectively for Kuka and for the operator. In this case, the result would be a JSON Array containing all the information requested, as shown below. Moreover, the initial state of each task is "created".



```
[
▼ {
    "det_short_id": "10132156G",
    "description": "COPERCHIO SEPARATORE OLIO COMPLETO",
    "error_time": "2020-08-04T16:50:24.000Z",
  ▼ "stop_time": null,
    "task_descr": "KIT motore 42100",
    "task_status_id": 2,
    "task_id": 1,
  ▼ "start_time": null,
    "agv_id": 1
  },
▼ {
    "det_short_id": "10132161F",
    "description": "COPERCHIO SEPARATORE OLIO COMPLETO",
  ▼ "error_time": null,
  ▼ "stop_time": null,
    "task_descr": "KIT motore 42100",
    "task_status_id": 2,
    "task_id": 2,
  ▼ "start_time": null,
    "agv_id": 1
  },
▼ {
    "det_short_id": "10132182F",
    "description": "** COPERCHIO SEPARATORE OLIO COMPLETO(A0",
    "error_time": "2020-08-05T17:36:17.000Z",
  ▼ "stop_time": null,
    "task_descr": "KIT motore 42100",
```

(a) Kuka Tasks



```
[
▼ {
    "task_id": 16,
    "det_short_id": "11092133F",
    "task_descr": "KIT motore 42100",
    "start_time": "2020-07-17T20:27:21.000Z",
  ▼ "error_time": null,
  ▼ "stop_time": null,
  ▼ "task_status_id": null
  },
▼ {
    "task_id": 17,
    "det_short_id": "11092141F",
    "task_descr": "KIT motore 42100",
    "start_time": "2020-07-17T20:30:59.000Z",
  ▼ "error_time": null,
  ▼ "stop_time": null,
  ▼ "task_status_id": null
  },
▼ {
    "task_id": 18,
    "det_short_id": "11212035G",
    "task_descr": "KIT motore 42100",
    "start_time": "2020-07-17T20:31:00.000Z",
  ▼ "error_time": null,
  ▼ "stop_time": null,
  ▼ "task_status_id": null
  },
▼ {
```

(b) Operator Tasks

Figure 4.11: Main Actors Tasks

As shown in picture 4.9, there are 5 branches.
Starting from the second one and the third one, they represent the logic related to the items in the shelf.
Those blocks, in fact, are agent centric, meaning that the main actor is the industrial container that is being positioned in the shelf or taken from it and it is the industrial container that passes through those blocks.
So, once items are injected according to the number expected (the number received in the initial JSON array at the main startup) they store themselves in a free position of the corresponding shelf, automatically.

Once positioned, they are sorted according to the pallet rack where they got, because one of the requisites of this UC is that items need to be collected sequentially (and not randomically) starting from the position where operator and kuka are at the beginning of the simulation. Hence, items are picked so that the first pallet rack (meaning the nearest to the related picking actor) is emptied, than the second one and so on.
As for the picking phase, some more details are needed in order to make the simulation more understandable.
Items being picked are, in fact, added to the corresponding cart (colors are modified dynamically in order to get at a glance that the item has been positioned in it) and in the meanwhile the picking time is measured both for worker and for Kuka.
Furthermore, in order to inform the remote operator about the evolution and the state of this work area, for each picked industrial container by Kuka, a GET request is sent to the following API in case the pick action went correctly:

```
http://icowms.cloud.reply.eu/Details/updateStatusOK?task_id="+kuka_remaining_items
```

(where "kuka_remaining_items" represents the identifier of the task in question), in order to keep track of the percentage of tasks completed in a real time mode, to visualize and possibly store some information about the downtime, etc...

Otherwise, if the picking did not go as it should have, the API contacted which will result in a dialog notification for the UI of the remote operator is the following:

```
http://icowms.cloud.reply.eu/Details/updateStatusEr?task_id="+kuka_remaining_items
```

and the corresponding task state would be set to "failed".
In case some error occurred in the picking phase (meaning that the "generate error" button has been clicked to simulate an anomaly), the operator needs to continue the execution of his tasks, while Kuka needs to stop until some corrective action has been performed by the remote operator.
So, another thread is created to keep listening to an API which returns the current

state of the erroneous task.
The method of the clientRestAPI called is listenService and the API is the following:

`http://icowms.cloud.reply.eu/Details/getTaskStatus?task_id="+taskId`

By means of a polling mechanism, every 5 seconds (this number is customizable as it has been stored inside a variable in the clientRestAPI class), the state of the task is obtained in order to know whether the simulation can go on or not.
Kuka will continue its execution if and only if the state of the task is "completed". Otherwise, if the state was "pending" (the remote operator decided to postpone the notification and take it on later) or still "failed", Kuka would stay fixed in the position where it stopped.

Once the two main actors finished their tasks, they become pedestrians (fourth and fifth branches), they position in the transshipment marker and the operator proceeds with the transfer of the packets from Kuka to the kit cart (whose actions are visible in the first branch).
Of course if one of the two actors finished before the other, it would wait until the other one completed its job, too.
Once the final kit is ready to go, an AGV comes from the pass and finally takes the cart outside.

The actors can now go back to their starting position waiting for a new order to be fulfilled, whenever the "Inject Again" button is clicked and still there is another scheduled order pending to be accomplished.

Figure 4.12 shows the situation of the blocks when an order has been completed.
This window is useful both during the execution to actually see which agents are passing through the blocks and at the end of it, just to check that everything went as expected.
The number at the top of a block represents the number of agents still in it, while the number at its left represents the agents that entered that block and the number at its right shows how many agents flowed out from it.
Looking at the blocks, in fact, the number of agents that went in is the same with respect to the number that came out and, due to the fact that one order has just ended up, the number on top is always 0, because each agent has been processed and sinked.

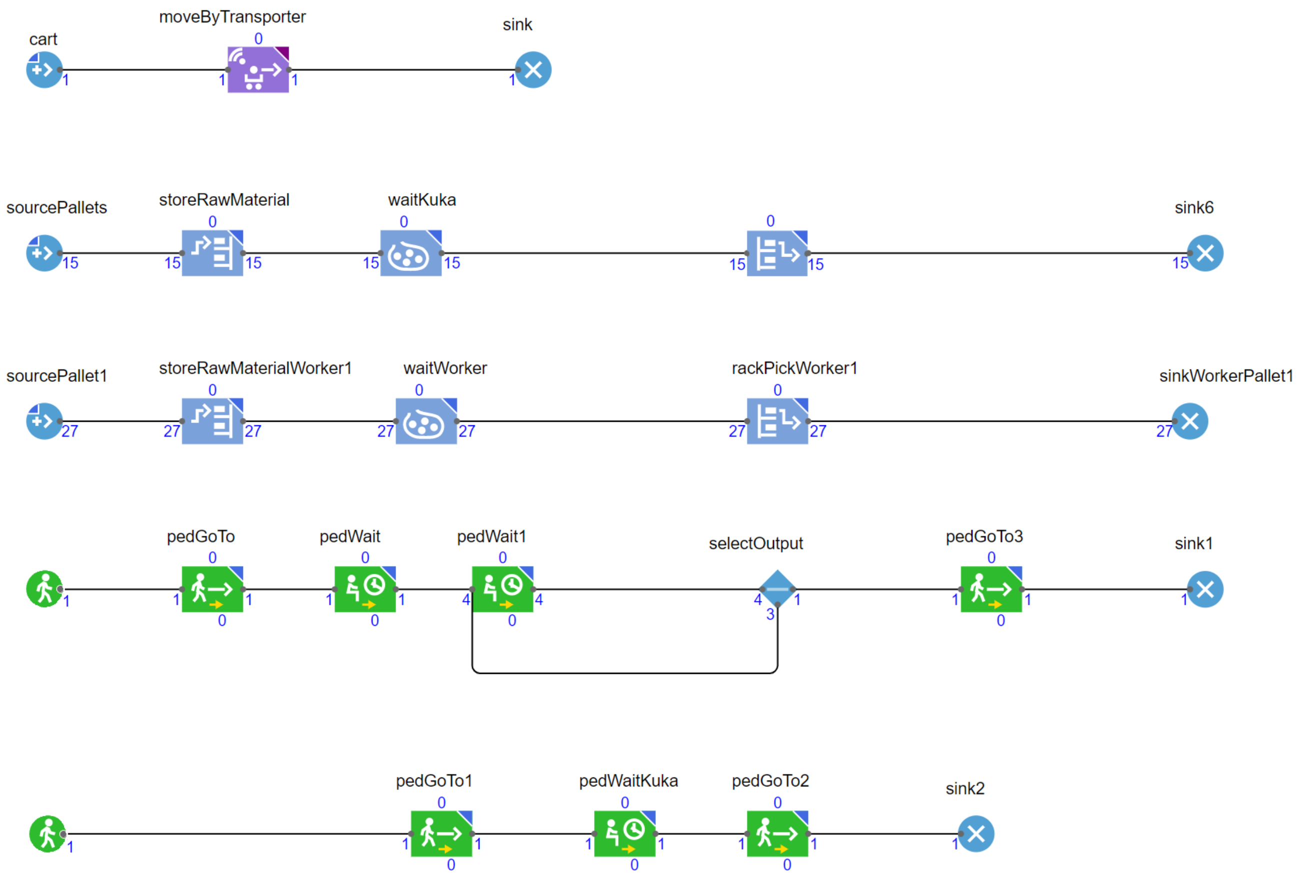


Figure 4.12: Logic after simulation

4.5 Statistics

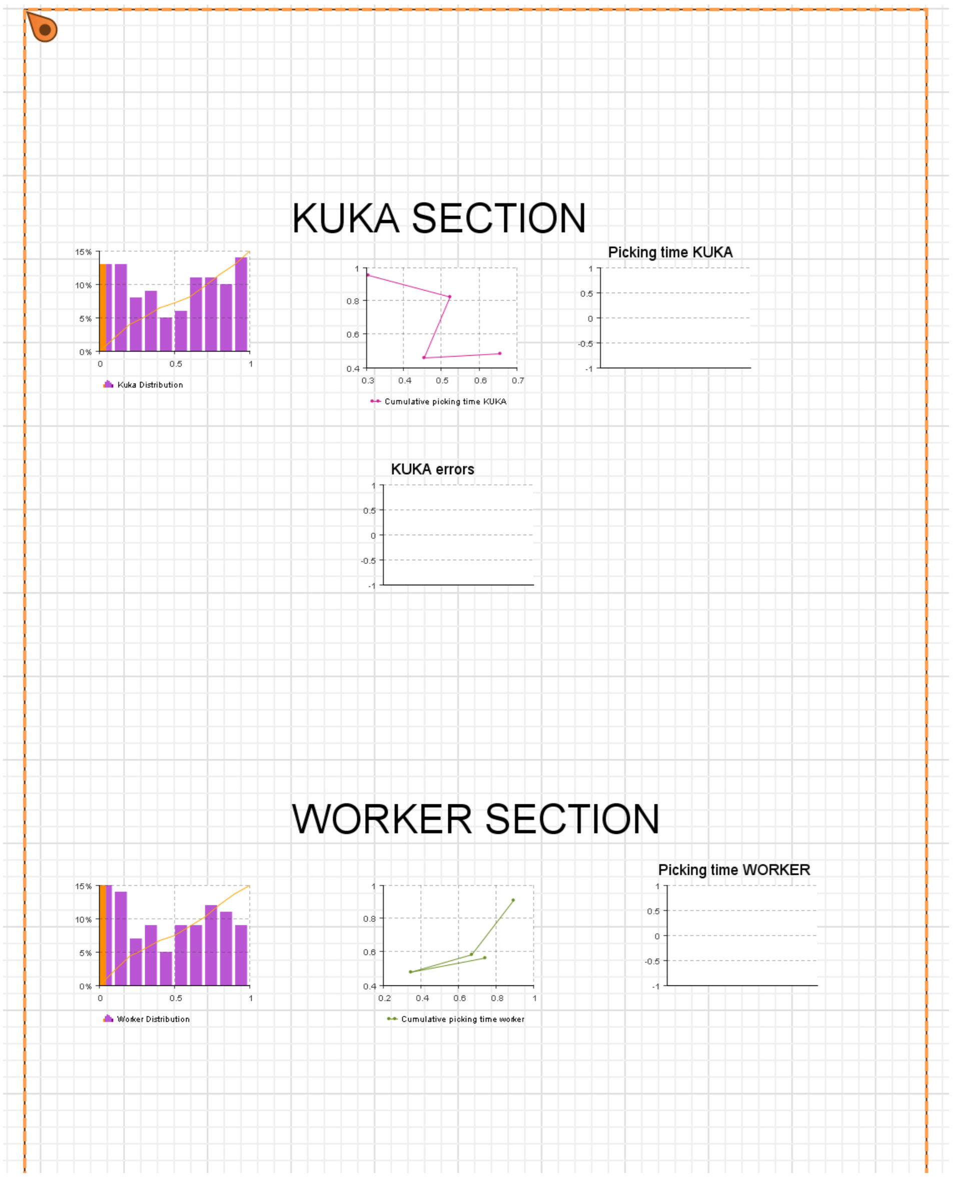


Figure 4.13: Statistics

The statistics phase is when all the important measures are collected to generate some graphics that could help in understanding more deeply what is happening to the simulation and this could lead to important insights.
As showed in picture 4.13 the statistics section is divided in two main groups, according to the main actors of this Use Case: Kuka and Worker.

4.5.1 Kuka statistics

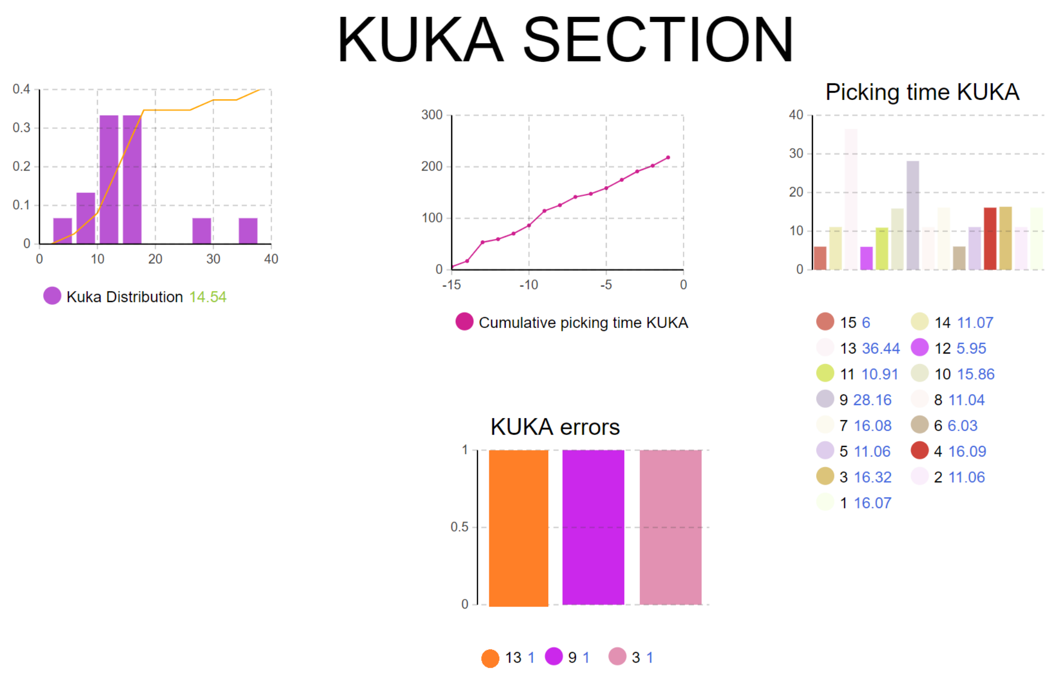


Figure 4.14: KUKA Statistics after simulation

The first graph is the Probability Density Function (PDF) of the time consumed to pick an industrial container and it is represented exploiting an histogram.
The values displayed in the graphics are calculated by taking the difference between the time when a pick is completed and the time when the resource to be picked is available (start time), for both the worker and Kuka.
As shown in figure 4.14, once one order has been completed the higher probability about the picking time corresponds to a value of 14.54 seconds for each item.
This value is consistent with the values for each single pick visible in one of the graphs that will be explained later on.

The second graph is a time plot, representing the samples of the dataset acquired during the simulations. On the X-axis there is the task id, while on the Y-axis the cumulative picking time value is visible.
The value sets to about 207.15 seconds, which is the total time Kuka spends for completing the "picking phase".
This could be one of the possible values to be optimized, in fact it could be reasonable thinking of reducing it, in order to lead to a faster overall time to complete an order.

However this will be discussed in the results chapter.

The third one is a bar chart showing the actual time spent to go pick something on the shelf. As this time is calculated as the difference between the task completion time and the starting one just the deltas are depicted and not the cumulative time. Each value is of course displayed with a different color in order not to get confused when reading it and to clearly understand what is its purpose.
This kind of information is very useful, in fact, starting from these values it is possible to see which task was the slowest or the faster to complete.
Moreover, by taking a look at it, the other statistics become more comprehensible. It is reasonable to infer, in fact, that the PDF provides 14.54 as the value with the highest relative likelihood in case another pick happened because the single values for the picking time mirror this value.
What could be interesting to point out is also that some tasks have higher picking values with respect to the other ones.
Why do we have this behavior?
Actually it could depend on the fact that once the cobot has emptied a specific pallet rack it has to move towards the next one and this requires more time for completing a certain task because the moving time and the actual time spent for the pick will contribute to increase the total time. On the contrary, being already positioned in the correct picking node, proceeding to complete the same pallet rack where the previous picked item was, should be faster than the other case just presented.
Also errors contribute to raise the time-to-pick value because in case of issues, Kuka is forced to wait until some action is performed, freeing it.
Another important factor could be the delay set at the beginning of the simulation which impacts tasks slightly differently. In fact some items are positioned, in terms of height, farther from the point where the pick takes place and this leads to an higher picking time with respect to items that are nearest to the height required by Kuka to pick them.

The last graph is representing, by means of a bar chart, the tasks where some errors occurred.
In this specific simulation the errors were generated to tasks 3, 9 and 13.
This is also why packets 3, 9 and 13 have higher picking time with respect to the other ones.
But why do they have different times if they were all affected by the same error?
Because the error required some time to be fixed and this amount of time was evidently different for each of them or simply because maybe the fix action was performed nearer or farther in time with respect to the 5 seconds polling to check whether the error is still present or if it has been solved.

It is important to point out that starting from the measures taken, it could be useful, knowing the total time of autonomy of Kuka, to get the remaining time before charging or even to calculate the idle time to try to optimize it.
Furthermore, it could be very useful starting from these graphics to get the saturation time, meaning the comparison between the time in which Kuka is actually working with the time in which it is waiting (so it does not produce any value), also for different missions.

4.5.2 Worker statistics

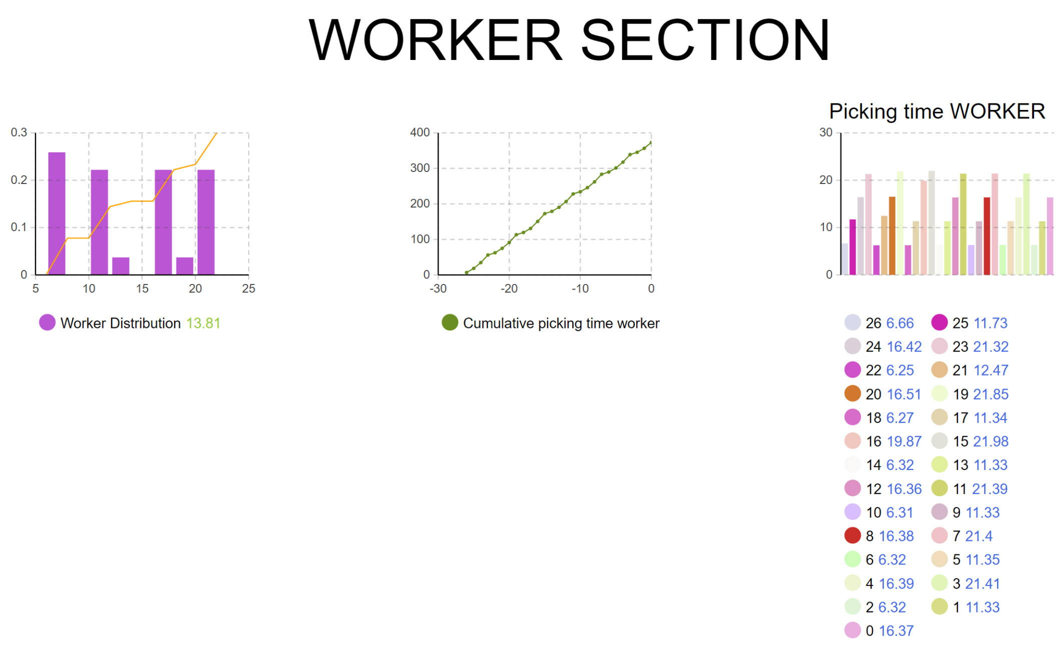


Figure 4.15: Worker Statistics after simulation

As for the worker, it is possible to see that the PDF is not influenced by possible errors happening during the simulation, also the single picking time values are quite uniform between each other, apart from some of them that could be ascribable to the same causes described in the previous subsection.
The cumulative time settles to a higher value with respect to the one shown in the Kuka section and this is due both to the fact that the operator picked more items and the fact that the operator moves slower than the cobot.

Chapter 5

Web Application

A web application has been built in order to simulate what the remote operator will face, once ICOSAF will reach its end, and in order to keep track of all the interfaces needed for the different use cases that will be implemented.

The first requirement for the User Interface part was to design it in such a way to be ready for all the Use Cases dealt in the project so, first of all a screen with a 4 button choice has been created in order to let operators move through the user interfaces of the various Use Cases (as can be seen in figure 5.1).
For each of them, an explicative image representing the corresponding UC has been chosen and inserted inside, so that at a first glance it is clear what is the purpose of the Use Case.

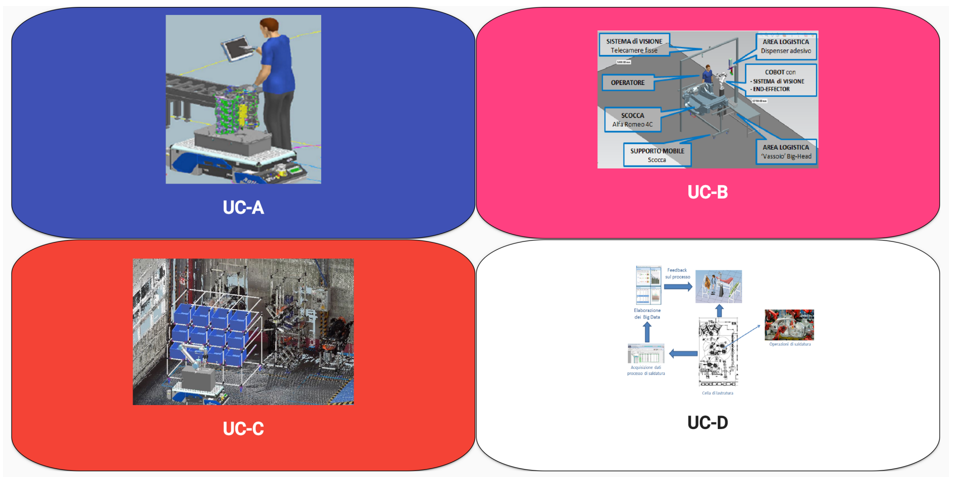

UC-A
SISTEMA di VISIONE Telecamere fisse
AREA LOGISTICA Dispenser adesivo
OPERATORE
COBOT con - SISTEMA di VISIONE - END-EFFECTOR
SCOCCA Alfa Romeo 4C
SUPPORTO MOBILE Scocca
AREA LOGISTICA "Vassoio" Big-Head
UC-B
UC-C
UC-D

Figure 5.1: Use Case Choice

By clicking the UC-C button a new page opens up with a dashboard (figure 5.2) that has been implemented, on the basis of the suggestions given by the web designer of the work group.
The dashboard is able to give a general overview about the state of each work area and each AGV in it, some graphics that show useful statistics and some commands (that will be better explained later) to interact with the simulated machines.

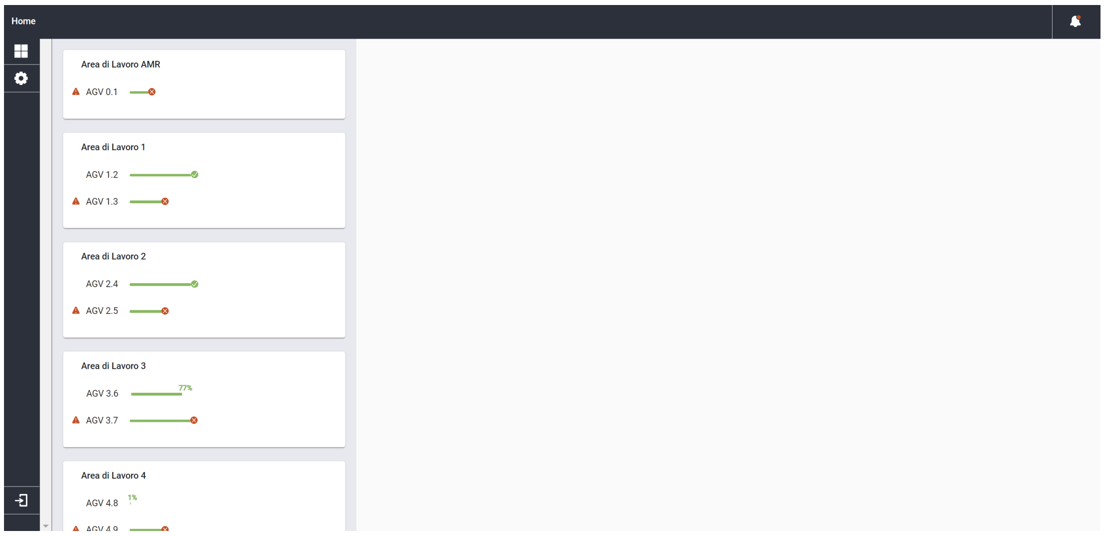


Figure 5.2: Dashboard

By taking a look at the dashboard (figure 5.2), it is possible to notice that there is a navbar which contains a button that will implement a notification list mechanism (top right corner) so that the remote operator can always check for previously received notifications.

On the left, instead, there is a sidenav with three main buttons representing the settings page (its functions will be implemented whenever there will be a clear idea about the content to be shown on click), the dashboard page itself and the login/logout button which will open the corresponding dialog.
Why should it be important to implement a login mechanism?
It is recommended to have, in case the operator needs to take important decisions, for example in correspondence of an error happened to Kuka while it was picking some items.
Furthermore, there could be some very important tasks that should be performed only if the person connecting to this web application was in charge of executing the operations that are being done, in fact, it is needed to insert a username and a password corresponding to a person who has the rights to perform the requested task.

In order to have a secure environment, not only an HTTPS communication has been implemented, but the credentials are encrypted by means of Crypto-JS (which is a javascript library of crypto standard) client-side and then decrypted server side to actually check whether they are correct or not.
The encryption standard that has been used is AES-256, with a 256 bit key generated starting from a passphrase.

In case the credential inserted were correct, the user would be authenticated and, if among the authorizations of this user there was at least one which attesting that this action (connecting to the web application with remote operator rights) could actually be performed, the user id of the corresponding operator would be sent back by the server, otherwise an error message would be received from it.

The most important part of the page is the one that occupies the most of it: the central one.
Here there is a list of vertical cards representing a work area each. How many work areas?
This information can be obtained by means of a GET to an API and then the number of AGVs present in each of them can be obtained in an analogous way.
Anyway this has to be implemented yet because project partners need to define whether this page should represent the information of only one work area (the simulated one) or, possibly, many of them.

However, it is clear that the focus of the dashboard is to let the remote operator know if there is something wrong that needs to be fixed somehow. In fact, colors and icons are properly used to catch the attention of the operator to the situations where something did not go as expected or, in general, where the attention of a remote user should be oriented.

Hence each work area, represented as a card, contains one or more AGVs, whose state is immediately visible at a glance.
The work area we will focus on will be Work Area AMR, whose name stands for collaborative Autonomous Mobile Robots for flexible logistics.

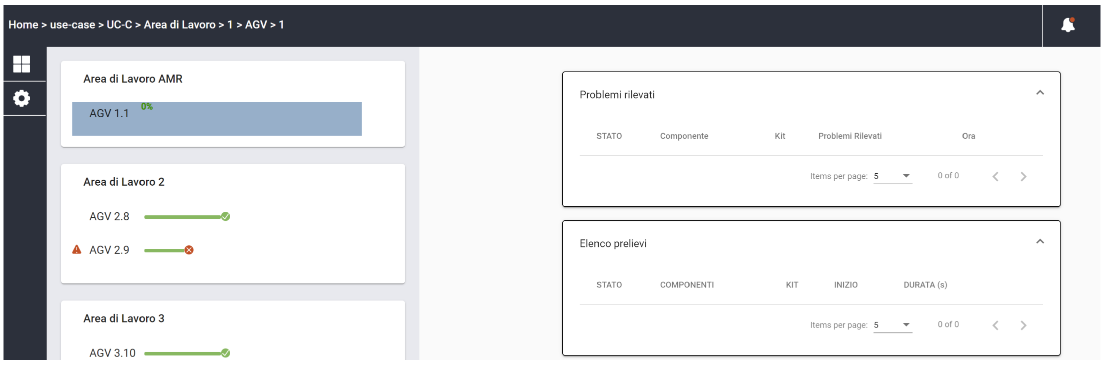


Figure 5.3: Dashboard percentage detail

In Work Area AMR (*"Area di Lavoro 1"*), for example, it is possible to notice in fact (figure 5.3) that AGV 1.1 has not picked any of its item yet, in fact, the completion percentage is still 0% but no error has occurred, while in work area 2, AGV 2.8 is proceeding correctly as well, but AGV 2.9 has encountered some issue. By clicking on the corresponding line some more information is shown, as visible in figure 5.4.

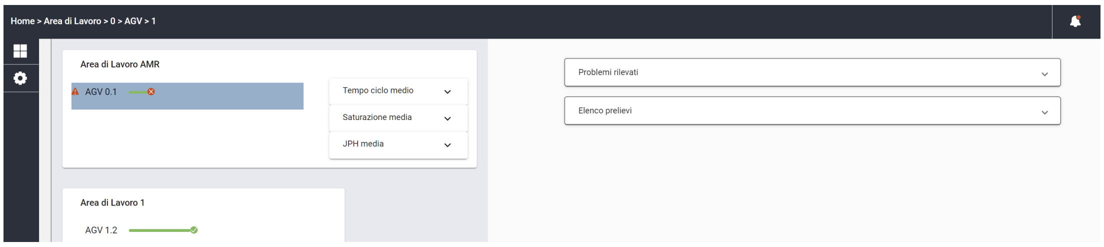
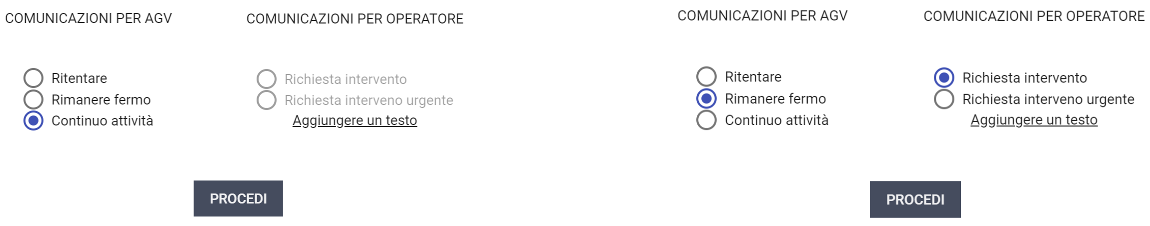


Figure 5.4: AGV details

Two expansion panels appear in a specific order. First of all there is the Errors panel (*"Problemi Rilevati"*) which will contain possible errors happened in the simulated environment during the picking phase of some items and will be populated as soon as an updateStaturErr is received by means of a simple click on the "Generate error" button of the AnyLogic 3D window.
The corresponding task will be described with: the status of the item in question, its ID, the kit it is part of, the kind of problem raised (*"Tipologia Problema"*), the time in which it happened and a button to solve this problem remotely (*"Risolvi"*). By clicking on this button a menu appears showing different options.
There are two kinds of communication that can happen: the one towards the collaborative robot (*"Comunicazioni Robot"*) and the one towards the operator in the shop floor (*"Comunicazioni Operatore"*).

The available options for communicating with the simulated Kuka are: Retry (*"Ritentare"*), Go on ((*"Continua attività"*), Stop and Wait (*"Rimanere fermo"*). In case this last option has been selected, the communication channel towards the operator is automatically opened, selecting one of its options (this could simulate the fact that the on field operator intervention is needed to solve this kind of issue). Figure 5.5 summarizes the mode of operation of the two radio groups.


COMUNICAZIONI PER AGV

Ritentare
Rimanere fermo
[✓ Continuo attività]

COMUNICAZIONI PER OPERATORE

Richiesta intervento
Richiesta interveno urgente
Aggiungere un testo

PROCEDI

COMUNICAZIONI PER AGV

Ritentare
[✓ Rimanere fermo]
Continuo attività

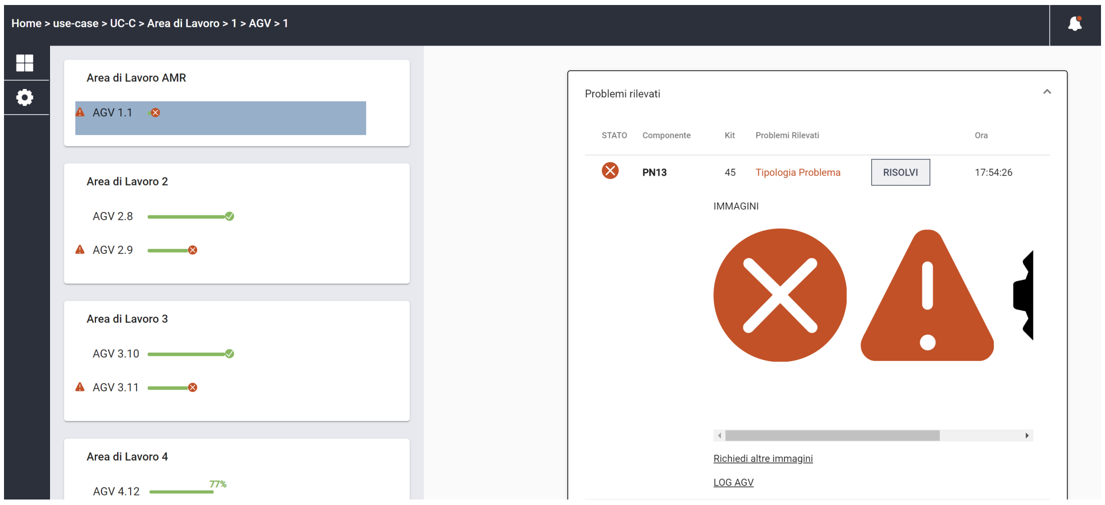
COMUNICAZIONI PER OPERATORE

[✓ Richiesta intervento]
Richiesta interveno urgente
Aggiungere un testo

PROCEDI

Figure 5.5: Communicating with Radio Groups

If the "Go on" option is selected, there is no need to communicate with the on field operator (therefore those options are disabled) and when the proceed button is clicked, the updateStatusOK API is called and the simulated cobot goes on and finishes its job.
With a simple click on the task error line, instead, it is possible to get some more information about it, such as images or the log of the AGV, as displayed in figure 5.6 that for the moment are simply hard coded.


Home > use-case > UC-C > Area di Lavoro > 1 > AGV > 1

Area di Lavoro AMR
AGV 1.1

Area di Lavoro 2
AGV 2.8
AGV 2.9

Area di Lavoro 3
AGV 3.10
AGV 3.11

Area di Lavoro 4
AGV 4.12 77%

Problemi rilevati

STATO	Componente	Kit	Problemi Rilevati		Ora
⊗	PN13	45	Tipologia Problema	RISOLVI	17:54:26

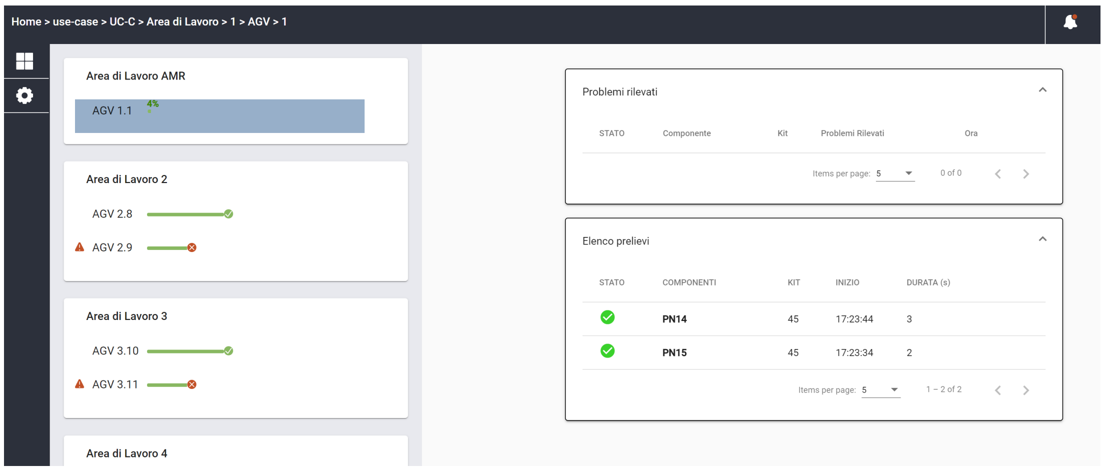
IMMAGINI

Richiedi altre immagini
LOG AGV

Figure 5.6: Problem details

The second expansion panel *"Elenco Prelievi"* represents a sort of history of the components that have been successfully processed and will be populated as soon

as an updateStatusOK is received or an error has been correctly solved.
In this case also the delay (meaning the duration of the picking phase related to that task) is shown, together with the time in which the task was started. Figure 5.7 shows packets 15 and 14 that have been correctly picked, but even though they belong to the same kit they have different delays and this is normal due to the fact that one item could be more heavy or more distant with respect to the position of the operator in the shop floor.
Also the percentage visible in the left side has changed according to the ratio between the correctly processed packets and the total number that need to be picked.


Home > use-case > UC-C > Area di Lavoro > 1 > AGV > 1

Area di Lavoro AMR
AGV 1.1 4%

Area di Lavoro 2
AGV 2.8
AGV 2.9

Area di Lavoro 3
AGV 3.10
AGV 3.11

Area di Lavoro 4

Problemi rilevati

STATO	Componente	Kit	Problemi Rilevati	Ora

Items per page: 5 0 of 0

Elenco prelievi

STATO	COMPONENTI	KIT	INIZIO	DURATA (s)
✓	**PN14**	45	17:23:44	3
✓	**PN15**	45	17:23:34	2

Items per page: 5 1 – 2 of 2

Figure 5.7: Successful picks

Why do some task appear with different task symbols? What do they stand for? According to the state a task may belong, different icons have been chosen (shown in figures 5.8, 5.9 and 5.10).

Figure 5.8: Status failed

Figure 5.9: Status success

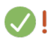


Figure 5.10: Status error solved

As for the work area itself, it is possible to have a general overview apart from seeing the situation of all the AGVs working in it.
In fact, some statistics can be obtained by clicking on the corresponding button, showing whenever the work area is clicked, like *"Tempo di ciclo medio"*, *"Saturazione media"* or *"JPH media"* to get respectively the mean cycle time, the mean saturation and the mean JPH.

This part still needs to be completed adding meaningful data to graphics (figure 5.11) and will be done as soon as it will be clear what data needs to be displayed.

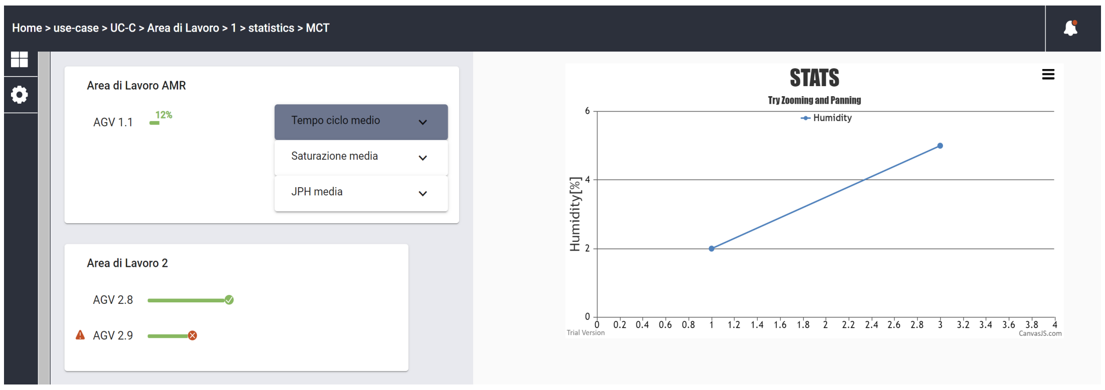


Figure 5.11: Web Application Statistics

In order to get change notifications HTML5 Server-Sent Events (SSEs) is used when the state of Digital Twins change.
Server-Sent Events (SSE) is a server push technology that enables a client to receive automatic updates from a server via HTTP connection.
The Server-Sent Events EventSource API is standardized as part of HTML5 by the W3C [31].

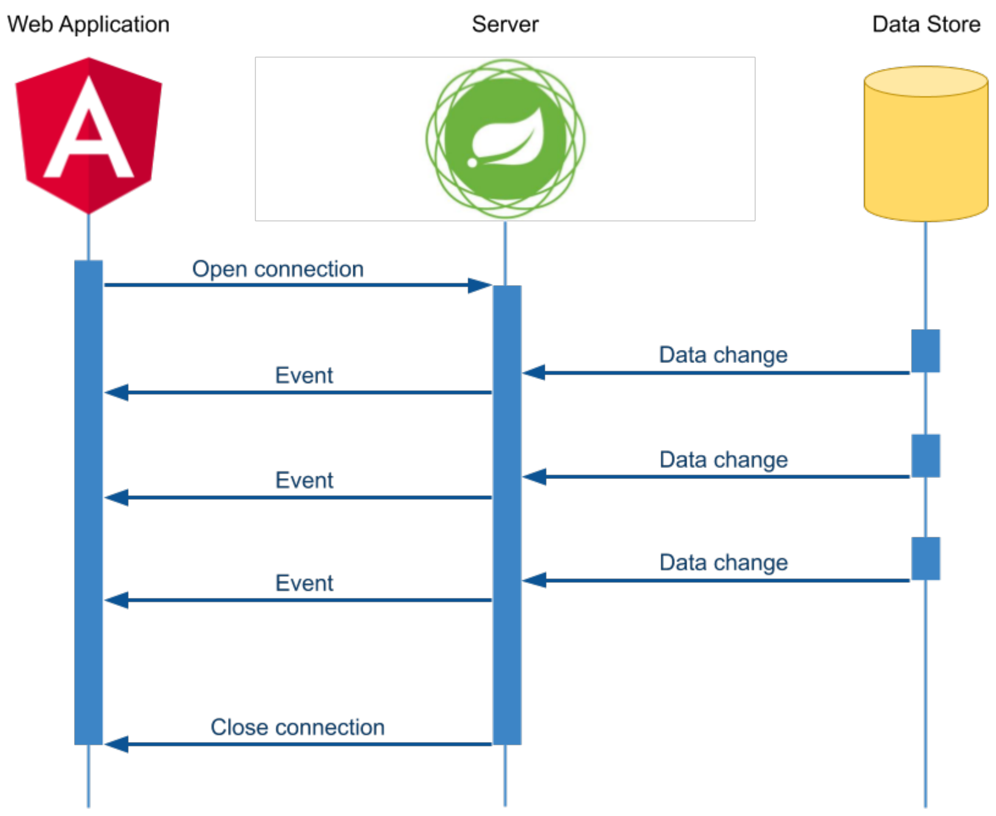
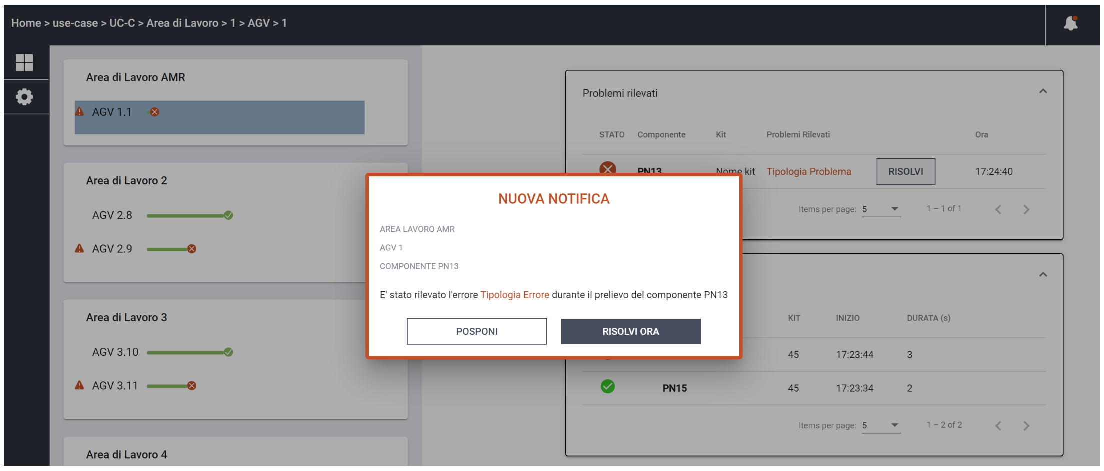


Figure 5.12: Server Sent Events [25]

The benefit of this mode of operation, contrary to the Web Socket channel, is that it is easier to open a SSE connection from the client than a Web Socket and moreover the events sent back from the backend have the same JSON structure as the HTTP API on which they are invoked.
Server-Sent Events allow a client (e.g. a web page in a browser) to get automatically data updates from a server.
This SSE client node sends a single HTTP request to the SSE server, and subscribes for all events at the server. As soon as an event occurs at the SSE server (when some stored data in a DB changes for example), the server will send the data of the event to this client. This way data streaming is accomplished.

Whenever an error has been raised by means of a click on the ”Generate error” button present in the simulated 3D environment, a notification pops up showing two possible actions corresponding to Postpone and Solve Now.

As can be seen in figure 5.13 an error has been generated (by means of the Any-Logic simulation 3D window) by Kuka, called AGV 1, to task number 13 in work area AMR.



Home > use-case > UC-C > Area di Lavoro > 1 > AGV > 1

Area di Lavoro AMR
AGV 1.1

Area di Lavoro 2
AGV 2.8
AGV 2.9

Area di Lavoro 3
AGV 3.10
AGV 3.11

Area di Lavoro 4

Problemi rilevati

STATO	Componente	Kit	Problemi Rilevati		Ora
	PN13	Nome kit	Tipologia Problema	RISOLVI	17:24:40

Items per page: 5 1 – 1 of 1

		KIT	INIZIO	DURATA (s)
		45	17:23:44	3
✓	PN15	45	17:23:34	2

Items per page: 5 1 – 2 of 2

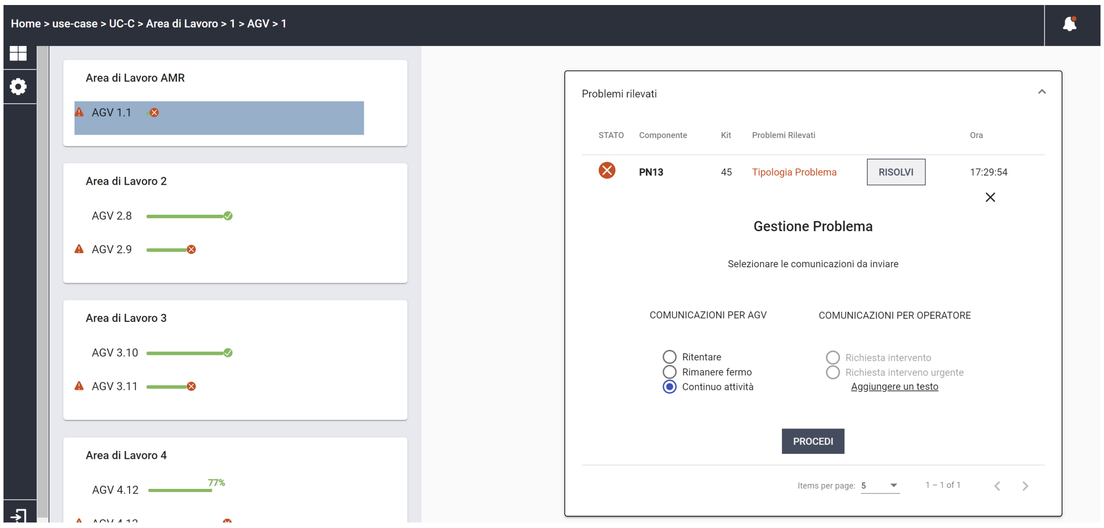
NUOVA NOTIFICA

AREA LAVORO AMR
AGV 1
COMPONENTE PN13

E' stato rilevato l'errore Tipologia Errore durante il prelievo del componente PN13

Figure 5.13: Notification

If the first button is clicked, the error is still there, Kuka keeps waiting until some corrective actions are taken, and the situation will be fixed in a second moment. If the second button is clicked instead, then the remote user (if already logged in) is redirected to the page visible in figure 5.14.



Home > use-case > UC-C > Area di Lavoro > 1 > AGV > 1

Area di Lavoro AMR
AGV 1.1

Area di Lavoro 2
AGV 2.8
AGV 2.9

Area di Lavoro 3
AGV 3.10
AGV 3.11

Area di Lavoro 4
AGV 4.12 77%

Problemi rilevati

STATO	Componente	Kit	Problemi Rilevati		Ora
	PN13	45	Tipologia Problema	RISOLVI	17:29:54

Gestione Problema

Selezionare le comunicazioni da inviare

COMUNICAZIONI PER AGV
Ritentare
Rimanere fermo
[✓ Continuo attività]

COMUNICAZIONI PER OPERATORE
Richiesta intervento
Richiesta interveno urgente
Aggiungere un testo

Items per page: 5 1 – 1 of 1

Figure 5.14: Solve Menu

Part III

Third Part

Chapter 6

Results

As it was explained at the beginning of this thesis work, the final objective was to integrate a Digital Twin and an HMI and furthermore to exploit the virtual system for further insights, optimizations and previsions.
The HMI that has been developed correctly responds to the initial request as it is possible not only to visualize data, but also to communicate with the simulation in both directions as if the system was real.
The HMI can now be validated with real time data and with realistic timings, since the Digital Twin is very similar to the collaborative space it simulates.
When the "Generate Error" button is clicked in the simulation environment, a dialog is shown to the remote operator in order to make the worker immediately aware of the kind of issue that arose.
Once the worker looked up at the details of this error (as explained in the dedicated section), it is possible to remedy by clicking one of the options shown.
Once the action has been taken, the communication goes the other way around and the UI informs the AnyLogic prototype of Kuka that it is possible to go ahead.
The simulated Kuka robot is finally resumed, but some other errors could happen and this mechanism can be repeated again and again.
By means of the simulation, therefore testing the HMI is not an issue anymore and it can be developed with much more functions (in case they were needed) than it has at the moment.
How to bring out the best in the developed Digital Twin?
It is possible now to optimize the performances of the collaboration between operator and the cobot by considering simple questions that can be answered only thanks to the simulation.

- Would it be useful to add another Kuka, trying to improve the speed of the overall Use Case related tasks?
  Certainly it would not. This is due to the fact that even though Kuka has a smaller shelf to manage when picking the items, the collaborative robot is much faster than the operator. In fact, on average, considering a number of items to pick,both for Kuka and the worker, which is very realistic [1] (as it was tried during the tests of the prototype), it emerges that Kuka always ends up waiting for the worker to terminate the picking phase (as can be seen in figure 6.1).
  Having two Kuka would result in having two cobots waiting rather than one and it would be more time consuming during the transshipment phase because the operator would have to empty two carts (one for each collaborative robot) rather than one.
  Furthermore, looking at the simulation, it could be important to consider the energy consumption of Kuka.
  If waiting for too long, in fact, in addition to the fact that it is not doing added value tasks, Kuka could consume a lot more than it should.
  Hence, the collaborative robot needs to turn to sleep mode in order to lower the battery expenditure. According to these considerations, increasing the number of collaborative robots, in this Use Case, would not be useful at all.

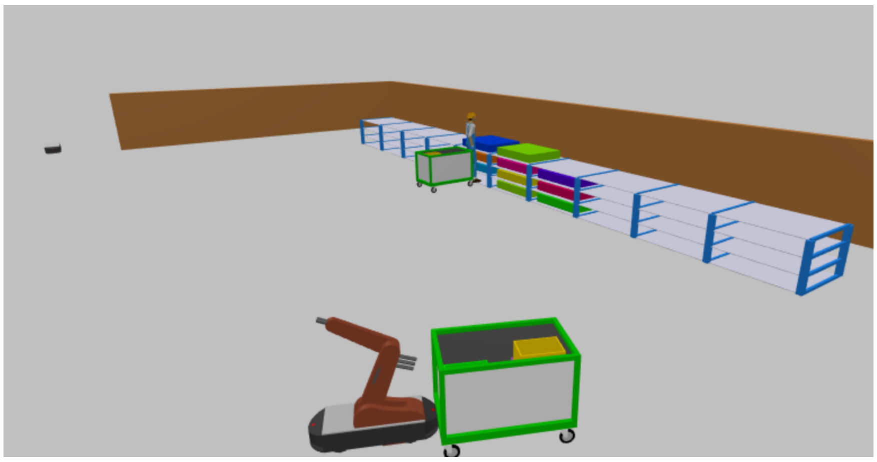


Figure 6.1: Kuka waiting for the operator to finish

- Would it be useful to remove the operator, exploiting two collaborative robots rather than one?
  Of course it would not. As it was clarified in the first steps of this thesis

[1]Realistic means that this number was taken into account as if the order was generated by a real WMS

work, industry 5.0 is human centric and this is because some human actions can not be substituted by any robot. In fact, for example, the check together with the preparation of the content of the final kit is delegated to the worker. Moreover, there could be some actions to perform for the worker in case some error occurred to Kuka.

- Would the performances be improved by increasing the number of AGVs?
  Also in this case, the answer is no. The AGV is always waiting a lot for the actors to terminate their picking phase, in fact, it needs to pick the final kit, then bring it to its final destination and eventually go back to the kitting area. The AGV has enough time to go and come back before another kit is ready to be picked (as shown in figure 6.2 where the AGV had time to go deliver the previous kit and it is now waiting for the next kit to be prepared).
  Therefore its energy consumption is already optimized for the added value tasks performed.

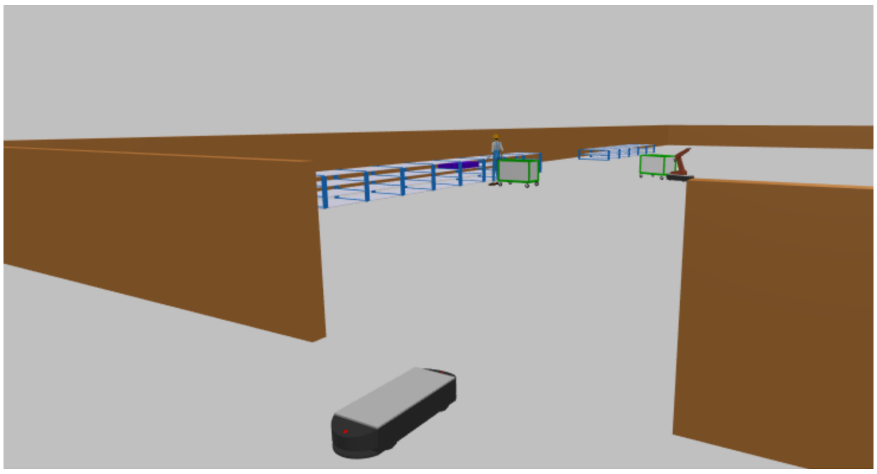


Figure 6.2: AGV waiting for the next kit

- Would it be useful to increase Kuka speed?
  No, it would not. As explained before, Kuka needs to wait for the operator to finish the picking phase. Increasing its speed would just result in less precision when picking or moving.

- The only thing that could be useful to consider is to have faster operators with respect to the one that have been virtualized. Anyway if on the one hand this could improve the overall performances (assuming that finding fast workers is a viable proposition), on the other hand it would result probably in less precision when picking and in a lower security, considering that this is a collaborative space.

Part IV

Fourth part

Chapter 7

Conclusions

In conclusion it is possible to assert that the thesis work has reached its main goal: developing a Digital Twin and an HMI and, subsequently, integrate them in order to have a working system that could be applied to a manufacturing company which exploits collaborative robots in the context of industry 4.0.
Digital Twin creation using AnyLogic allowed us, in this early stage of the project, to substitute the real system with a simulated one in order to enable the development and the testing for the HMI communication with the "real-time system" and, moreover, allowed the real system assessment from the performance point of view, before the real implementation in a real manufacturing environment happened.
At the beginning of this work, in fact, there were just some hints about how the use case analyzed should have worked, which included some parameters about the speed of the collaborative robots and the positions of the shelves, without having a proper representation of it, making it quite difficult also to understand how it would have been implemented.
Now it is possible not only to clearly see a realistic 3D realization of the collaborative space, but also to obtain all the insights and the potentialities promised by the Digital Twin technology.
Furthermore, the web application has been developed having always the final user in mind in order to make it usable, efficient and effective of course.
For what concerns further improvements, it is clear that the final work is still not finished and will be growing together with the project as it goes on, responding to the requests that may arise from FCA.
The next step, once the project has finished, could be to introduce some Machine Learning technique to individuate some common patterns, in order to get some results from an Artificial Intelligence, gaining some interesting insights from the data analyzed and obtained from the Digital Twin.

Bibliography

[1] *Access Your Control Systems on Mobile Devices.* URL: https://inductiveautomation.com/ignition/modules/mobile-scada-hmi.

[2] Majeed Ahmad. *Augmented Reality Hardware: Key to HMI.* URL: https://eu.mouser.com/applications/Augmented-Reality-Hardware-Key-to-HMI/.

[3] AnyLogic. *An introduction to digital twin development.* Tech. rep.

[4] Anylogic. *Anylogic Simulation Software.* URL: https://www.anylogic.com/.

[5] AnyLogic. *Simulation software comparison.* Tech. rep.

[6] AnyLogic. *What's new in AnyLogic? Benefits of the built-in database.* URL: https://www.anylogic.com/blog/what-s-new-in-anylogic-benefits-of-the-built-in-database/.

[7] Inductive Automation. *What is HMI?* URL: https://www.inductiveautomation.com/resources/article/what-is-hmi.

[8] Rosen R. Boschert S. "Mechatronic Futures". In: Springer, Cham, 2016. Chap. The Simulation Aspect.

[9] Capterra. *AnyLogic vs Matlab.* URL: https://www.capterra.com/simulation-software/compare/95940-125813/AnyLogic-vs-MATLAB.

[10] Deloitte. *Industry 4.0 and the digital twin.* URL: https://www2.deloitte.com/us/en/insights/focus/industry-4-0/digital-twin-technology-smart-factory.html#:~:text=The%20digital%20twin%20configuration%20of,Deloitte%27s%20approach%20to%20Industry%204.0..

[11] *Digital Twin Technology: Why Is It Important?* URL: https://www.sam-solutions.com/blog/digital-twin-technology-why-is-it-important/.

[12] General Electric. URL: https://www.ge.com/research/offering/digital-twin-creation.

[13] Engineering. *Digital Twin, A digital copy of reality that enables you to simulate and find answers in a risk free and secure environment.* URL: https://www.eng.it/resources/whitepaper/doc/digital-twin/digital-twin_whitepaper_en.pdf.

[14] *Front End Frameworks, Front-end frameworks and libraries, Rankings.* URL: https://2019.stateofjs.com/front-end-frameworks/.

[15] Marcia Gadbois and Jeff Payne. *Improve remote HMI and OIT access.* URL: https://www.controleng.com/articles/improve-remote-hmi-and-oit-access/.

[16] J.J. Garrett. *The Elements of User Experience.* New Riders, 2011.

[17] Direct Industry VirtualEXPO Group. *Robot mobile KMR iiwa.* URL: https://www.directindustry.it/prod/kuka-ag/product-17587-1714901.html.

[18] ICOSAF. *Document of Work.*

[19] *Impressions of Garrett's "Elements of UX.* URL: http://www.johnferrigan.com/blog/2014/8/28/impressions-of-garretts-elements-of-ux.

[20] Indiamart. *AGV - Automated Guided Vehicle.* URL: https://www.indiamart.com/proddetail/agv-automated-guided-vehicle-10989805612.html.

[21] Fraunhofer IPT. *Smart Glasses Guide makes it easier to choose data glasses.* URL: https://www.hannovermesse.de/en/news/news-articles/smart-glasses-guide-makes-it-easier-to-choose-data-glasses.

[22] Thorsten Krüger. *How Wearables revolutionize the Human Machine Interface in Industry.* URL: https://medium.com/workerbase/how-wearables-revolutionize-the-human-machine-interface-in-industry-2be987ca4cfe.

[23] KUKA. *KMR iiwa.* URL: https://www.kuka.com/it-it/prodotti-servizi/mobilit%C3%A0/robot-mobili/kmr-iiwa.

[24] J. Nielsen. *Usability Engineering.* 1994.

[25] OctoPerf. *FULLSTACK REACTIVE SERVER SENT EVENTS.* URL: https://octoperf.com/blog/2019/10/09/kraken-server-sent-events-reactive/.

[26] Head of industrial doctoral school di Eit Digital Roberto Saracco. URL: https://www.agendadigitale.eu/industry-4-0/industry-4-0-modello-digital-twin-migliora-sviluppo-prodotti/.

[27] *Simulation Software for Every Business Challenge.* URL: https://www.anylogic.com/features/.

[28] Cinzia Testa. *Carico cognitivo da iperconnesione nell'era del coronavirus.* URL: https://www.quotidianosociale.it/carico-cognitivo-da-iperconnesione-nellera-del-coronavirus/.

[29] Johanna Turesson. "Remote Control of an AGV, Development of a User Interface for Automated Vehicles". MA thesis. Chalmers University of Technology, Gothenburg, Sweden, 2016.

[30] *What about Angular? Angular JS history through the years (2009-2019)*. URL: https://www.ryadel.com/en/angular-angularjs-history-through-years-2009-2019/.

[31] Wikipedia. *Server-Sent Events*. URL: https://en.wikipedia.org/wiki/Server-sent_events.

Ringraziamenti

A conclusione del mio percorso di studi , mi sembra opportuno rivolgere i miei più sentiti ringraziamenti a tutte le persone che, direttamente o indirettamente, hanno contribuito e reso possibile il raggiungimento di questo importante traguardo.

Desidero ringraziare i miei genitori, che con molti sacrifici, sia dal punto di vista economico sia, soprattutto, affettivo mi hanno permesso di studiare in una città diversa dalla mia e di raggiungere questo traguardo, rimanendo al mio fianco nei momenti più complessi, facendomi sentire sempre a casa con il loro infinito affetto. Non riuscirò mai ad esprimere appieno l'affetto che provo per entrambi, ma è a loro che devo tutto quello che sono diventato.

Ringrazio mio fratello Vincenzo, la mia spalla da una vita, figura fondamentale per me. Fin da quando eravamo bambini, accompagna i miei passi con umiltà e risolutezza. Non avrei potuto desiderare persona migliore per starmi vicino, giorno dopo giorno.

Ringrazio Deborah che, nonostante la lontananza fisica, ha sempre partecipato ad ogni mia battaglia, sostenendomi ed incoraggiandomi. La sua presenza, il suo affetto ed i momenti che abbiamo condiviso mi hanno reso una persona migliore.

Ringrazio i miei zii, le mie nonne, i miei cugini e tutta la mia famiglia che rappresentano una parte indispensabile della mia vita e che mi hanno trasmesso i valori più importanti.

Ringrazio di cuore mio zio Gino (al quale è anche dedicata la tesi), che mi ha trasmesso con esuberanza e fermezza la vera essenza del termine "famiglia", che mi protegge e guida dall'alto ogni mia azione. Lo sento vicino ogni giorno e sono sicuro che starà sorridendo per il suo *nano ghiacciato* mentre suona *Samba pa ti.*

Un sentito ringraziamento va a mio zio Santino che con la sua dedizione, le sue conoscenze, i suoi consigli, affabilità e gentilezza mi ha sempre supportato e aiutato in qualunque momento ne avessi bisogno, diventando per me , inevitabilmente, una guida da seguire e, non solo, dal punto di vista professionale.

E, a questo punto, come non ringraziare in modo speciale i miei amici, soprattutto quelli di vecchia data, che, forse senza rendersene conto, mi hanno agevolato dandomi la carica necessaria per affrontare tutto con più semplicità e leggerezza.

Grazie alla *SbadaGang*: Walter, Marco e Giacomo con i quali ho stretto un forte legame che va oltre il semplice essere colleghi, assumendo la connotazione di veri amici.

Una menzione particolare va a tutti gli insegnanti, che con la loro presenza e competenza hanno contribuito a tracciare il mio percorso di vita. Un itinerario che si snoda dalla scuola elementare fino agli studi universitari. In particolare, a questo proposito, mi piace ricordare le mie maestre Annabella e Antonella, le professoresse Lo Giudice, Loreta, Lupò, Fleres, Latino, Manganaro, Muscolino, Chiusano, i professori Di Leo, Prestipino, Bernardi, Morisio.

Un ringraziamento speciale lo rivolgo a Liudmila Dobriakova, con la quale ho condiviso, sia le gioie che le tribolazioni di un itinerario non sempre facile, riconoscendo che senza il suo contributo questo percorso finalizzato alla realizzazione della tesi sarebbe stato senza dubbio più tortuoso e ostico.

Infine, non certo per importanza, ringrazio il Prof. Paolo Garza per essersi sempre dimostrato disponibile in veste di professore, prima , ed in quella di relatore, dopo, e ,soprattutto, per aver saputo indirizzarmi coniugando nelle sue azioni: attenzione e professionalità, ingredienti indispensabili, che hanno fatto sì che il lavoro procedesse nel migliore dei modi.