TÉCNICO LISBOA

Digital Twin of a Flexible Manufacturing System for Solution Preparation

Paulo Francisco Carreira Faria

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof. João Miguel da Costa Sousa
Prof. Susana Margarida da Silva Vieira

Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira

Supervisor: Prof. Susana Margarida da Silva Vieira

Membes of the Committee: Prof. João Manuel Gouveia de Figueiredo

December 2021

Acknowledgements

Firstly, I would like to say that I am grateful for the help of Tiago Coito and Miguel Martins, who were involved in the development of the prototype, for the guidance they gave me through every step of the development of this thesis, providing me with suggestions and feedback, as well as information about the Robot System.

I would also like to extend my gratitude to my supervisors, João Sousa and Susana Vieira, for introducing and providing me the opportunity to conduct my thesis on a topic that I find most interesting, as well as their insight and guidance.

A special thanks goes to my parents, for always being there for me, and helping me be the person I am today.

Abstract

In the last decades, there has been a necessity for systems capable of handling market changes and various personalized customer needs, with near mass production efficiency, defined as mass customization. Industry 5.0 further exposed this need for robust flexible systems, as well as a necessity of manufacturing systems that work in close cooperation with workers, taking advantage of the problem-solving capabilities and knowledge of the manufacturing process.

A solution for this necessity is to develop a flexible manufacturing system, capable of handling different customer requests and real-time decisions from operators. Which this thesis tackles by proposing a Digital Twin focused on the simulation of a Robot System for Solution Preparation, capable of making real-time scheduling decisions and forecasts by using a simulation model to test different resource configurations and customer requests, while allowing an operator to make changes in the processing time and order of some operations in real-time.

A mixed event simulation model was utilized to do resource forecasts, where real-time decisions were performed by using completely reactive scheduling with parallel tasks. Resource forecasts were utilized to know where the manufacturing system can be improved. It was shown that combining parallel tasking with parallel machines to key processes, utilizing heuristics that emphasize the shortest transportation time and increasing the robot’s speed, best impact the performance of the system, reducing overall completion time by 82%, when comparing single tasking and single machines. The simulation model also has an animated visualization window, for a deeper understanding of the system.

Keywords

Digital Twin, Mass Customization, Industry 5.0, Flexible Manufacturing System, Simulation

Resumo

Nas últimas décadas, tem havido uma necessidade de sistemas industriais capazes de lidar com alterações do mercado, e assim, com pedidos diferentes de clientes de maneira eficiente, que se define como personalização em massa. A visão da indústria 5.0 expôs mais essa necessidade de sistemas robustos flexíveis, bem como a de sistemas industriais capazes de trabalhar em cooperação mais próxima com trabalhadores, de modo a aproveitar a sua capacidade de resolver problemas e conhecimento dos processos.

Uma solução para este problema é desenvolver uma célula flexível de fabrico, capaz de lidar com diferentes pedidos de clientes e decisões em tempo real de trabalhadores. Esta tese aborda este tema, desenvolvendo um Gémeo Digital focado na simulação, de um Sistema Robótico de Preparação de Soluções, capaz de fazer escalonamento em tempo real e realizar previsões com diferentes pedidos de clientes, com a possibilidade de um operador alterar o tempo de alguns processos e a ordem de algumas operações em tempo real.

Um modelo de simulação foi desenvolvido para realizar previsões, recorrendo a escalonamento completamente reativo com tarefas em paralelo, para entender onde o sistema pode ser melhorado. Demonstrou-se que utilizar tarefas em paralelo, adicionar máquinas idênticas em processos chave, utilizar heurísticas que colocam em ênfase o menor tempo de transporte e aumentar a velocidade do robô, têm o melhor impacto na performance do sistema, reduzindo o makespan em 82%, em comparação com tarefas individuais. O modelo de simulação também permite a visualização com animação, para uma compreensão mais aprofundada do sistema.

Palavras-chave

Digital Twin, Personalização em Massa, Industria 5.0, Célula Flexível de Fabrico, Simulação

Contents

List of Tables

List of Figures

Acronyms

AI	Artificial Intelligence
BPMN	Business Process Modelling Notation
CSMT	Current Shortest Movement Time
DMS	Dedicated Manufacturing Systems
DES	Discrete Event Simulations
DTS	Discrete Time Simulations
ES	Entrance Storage
FS	Final Storage
FMS	Flexible Manufacturing Systems
FEL	Future Event List
LWR	Least Work Remaining
LPT	Longest Processing Time
RMS	Reconfigurable Manufacturing Systems
SMT	Shortest Movement Time
SPT	Shortest Processing Time
TAM	Time Advance Mechanism
W1	Workstation 1
W2	Workstation 2
W3	Workstation 3
W4	Workstation 4
W5	Workstation 5
W6	Workstation 6
W7	Workstation 7

Chapter 1

Introduction

During the last decades, there have been remarkable leaps in the areas of information technologies and digitalization, that lead to the creation of autonomous, self-regulated systems.

These developments increased the output of the industry by a large margin, but the growing variety and rising costumer demand for individual or custom-made products at lower costs, calls for the design and operation of systems capable of handling this increasing variety in products[1]. This need to deliver products and services that best meet individual customers' needs with near mass production efficiency, is defined as mass customization [1], [2].

There have been ways to deal with this necessity in the past, such as recurring to Reconfigurable Manufacturing Systems (RMS) in which machines, components, and material handling equipment can be added, removed, modified, or interchanged as needed to respond quickly to changing requirements [3], offering more variety compared to Dedicated Manufacturing Systems (DMS) as shown in figure 1.1.

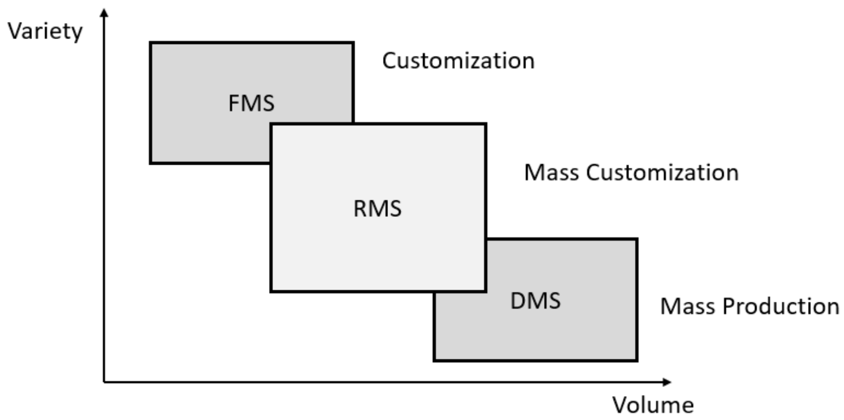

Figure 1.1: Relation between variety and volume of different manufacturing systems, based on [4].

Another way to deal with this issue was the use of Flexible Manufacturing Systems (FMS), in which, the main idea, is a system built *a priori* to deal with changes in the market demands, able to yield a wide range of products from a single base unit.

FMSs are better fit for high customization and low production volume, figure 1.1 shows the correlation between variety and volume of productions of different manufacturing systems [3].

1.1 Motivation

Depending on the necessity of variety in customization and volume optimized for a lower cost, in the past, enterprises usually applied these manufacturing systems, but there is still an even greater necessity for customization and the capability of offering more variants per model, and introducing new models faster, still constrained by the current technologies and the equipment of mass production operations [5].

This thesis addresses the need for mass customization and flexible robust systems, by developing a Digital Twin of a manufacturing system, capable of handling different requests and real-time changes from operators and clients, by making use of the new technological trends.

T. Coito et al. [6] first presented this manufacturing system as a case study, in the quality control laboratory of the pharmaceutical industry. The authors propose a platform that allows for the autonomous acquisition and management of personalized data in real-time for mass customization manufacturing environments, that supports the integration of dynamic Decision Support Systems.

1.1.1 Industry 4.0

Achieving flexibility and adaptability that can be defined as the production system's sensitivity to internal and external changes, is regarded as one of the most promising solutions over the last years, with the rise of Industry 4.0, defined by the German Federal Government as an emerging structure in manufacturing and logistics systems in the form of Cyber Physical Systems (CPS). CPS are systems of collaborating computational entities that are in connection with the surrounding physical world and its on-going processes, providing and using, at the same time, data-accessing and data-processing services available on the internet, intensively use the globally available communications network for an automated exchange of information and in which production and business processes are matched [7].

Industry 4.0 relies on concepts such as the Internet of Things (IoT), that describes a network of interconnected physical objects that share information given by sensors and software between each other, cloud based manufacturing which allows organization to easily store, update and apply information and smart manufacturing which takes advantage of powerful information and manufacturing technologies that enable flexibility in physical processes for the current dynamic and global market [8]–[10]. These pillars of Industry 4.0 transform production cells into a fully integrated, automated, and optimized production flow [7].

Mass customization is currently for the most part, enabled by Industry 4.0 technologies, including Internet connections between dealership ordering systems, supply chain systems, and even the robots on the car factory floor. But this is not enough, as costumers want more and more options for customization [11].

Unfortunately, these changes in the industry might have a negative effect on society itself, mainly on industry workers, as the development of technologies also puts in jeopardy their positions in the industry as changing roles and increased reliance on complex technologies will require new skills, meaning more profound changes in how the workforce is organized will present themselves, challenging industry workers traditional education life cycle of training, work and retirement [12].

Consensus is emerging that routine jobs with lower creative requirements are most at risk, while evidence suggests that non-routine manual labor is broadly unaffected as non-routine cognitive tasks have been complemented by computers. Frey and Osborne [13] estimated the probability of computerization for 702 detailed occupations. They concluded that 47% of all US employment is in a high-risk category [14].

1.1.2 Industry 5.0

The term was firstly introduced on December 1st, 2015 [15], Industry 5.0 is a future trend that prioritizes closer cooperation between man and autonomous machines, maximizing the efficiency of both counterparts, by taking advantage of the human mental, creative capabilities, in a way, returning the man to the “Center of the Universe” [16], [17].

In terms of technology, Industry 5.0 wants to come to grips with the promises of advanced digitalization, big data and artificial intelligence, while emphasizing the role these technologies can play to address new, emergent requirements in the industrial, societal and environmental landscape. This means using data and AI to increase production flexibility in times of disruption and rendering value chains more robust; it means deploying technology that adapts to the worker, rather than the other way around [12].

Industry 5.0 will change the definition of the word “robot”. Robots will not be only a programmable machine that can perform repetitive tasks but also will transform into an ideal human companion for some scenarios. Providing robotic productions with the human touch, the next industrial revolution will introduce the next generation of robot, commonly termed as Collaborative Robots (CoBot), that will already know, or quickly learn, what to do [17].

1.2 Digital Twin

The idea of a Digital Twin (DT) refers to a comprehensive physical and functional description of a component, product or system, which includes more or less all information which is considered useful, so that a digital entity of its own could be created, considered as a “twin”, that allows the exchanging of information between the real system and its digital counterpart .

The DT, it is not only useful to describe the behavior of a system, but also to derive solutions relevant for the real system, it evolves along the real system throughout its lifetime. The communication between these counterparts is facilitated by the use of IoT which increases the amount of available usable data, helping by providing information relevant for monitoring and decision-making process given by sensors, engineering data, behavioral descriptions and software, such as the status of the object, maintenance and reliability [18].

The Digital Twin when coupled with data analytics allows for real-time monitoring, rapid analysis, and real-time decisions [19], [20], allows companies to quickly detect problems in physical systems, increase the accuracy of their results and more efficiently produce better products [18].

The Digital Twin is part of the industry 4.0, and is an addition to simulation technology, that started in the 1960s to solve design problems [21], and it progresses according to figure 1.2.

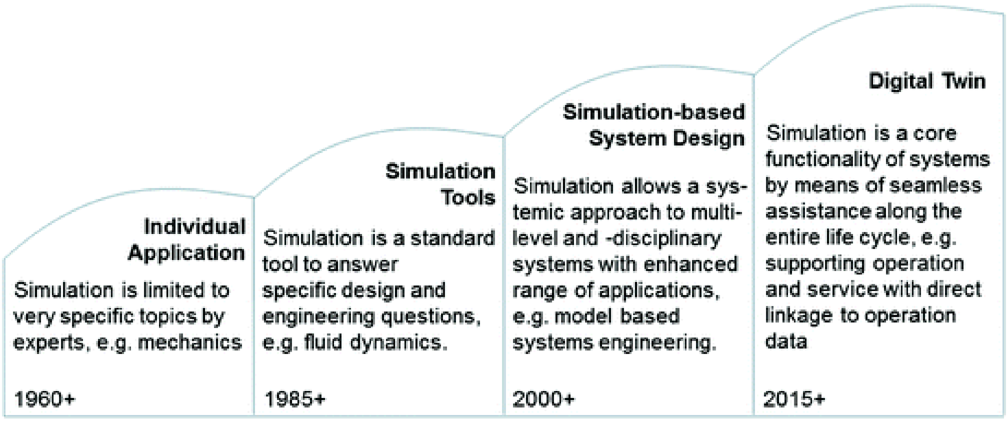

Figure 1. 2: Progress in simulation technology [21]

Simulation nowadays is the basis of a lot of design decisions, testing of new products and validation of both components and systems in many fields, with the arrival of the DT, constant integration of currently available data and knowledge from the real twin allows simulation models not only to be relevant in the design phase of products but also throughout the products lifetime [19].

It is with no surprise that the Digital Twin is a very attractive concept nowadays, according to a recent survey by Gartner [22], of the companies that are implementing IoT, 13% already use DTs, and 62% are in the process or plan to establish one.

The concept is not a new one, it was first introduced by Grieves in 2002 for the formation of a Product Lifecycle Management (PLM) center, initially called “Conceptual Ideal for PLM”, although it did not have the current name, it shared the elements of a DT, those being, the real space, the virtual space and the links for data flow between both spaces [20]. The term itself originated from the National Aeronautical Space Administration (NASA), which released a paper in 2012 called ‘‘The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles’’, in this paper, NASA defined the term as:

‘‘A Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin’’[23].

This is a general definition as there is still no consensus in what exactly a Digital Twin is, in both academia and industry there is often no distinction from general computing models and simulations [24]. For better understanding and to clear misconceptions, there are 3 levels of digital integration, as shown in figure 1.3, these are:

Digital model: A digital version of an already existing or planed physical object, there is no automatic information exchange between the real and digital version, this means, any changes in both physical and digital versions have no effect in the other counterpart, for example, plans for buildings, product designs and development.

Digital shadow: Digital representation of an object in which, communication only happens from the real model to the digital one, and so, a change in the physical model originates an automatic change in its digital counterpart, but not the other way around.

Digital twin: Data flow exists in both directions, so any changes in the physical object, originates a change in the digital object and vice versa [24].

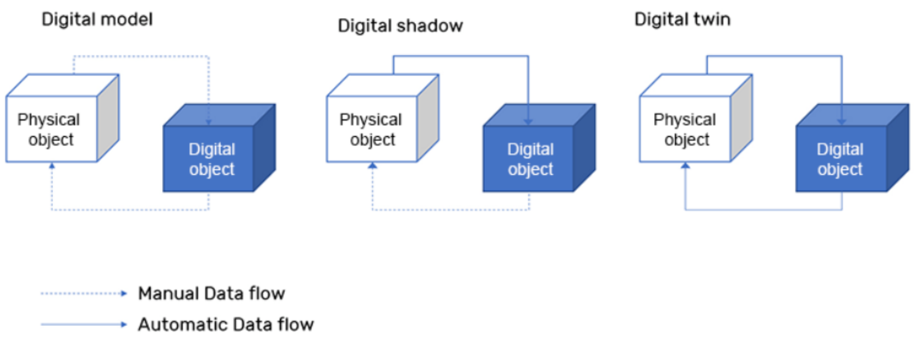

Figure 1. 3: Levels of digital integration [21]

1.2.1 Digital Twin Applications

The Digital Twin can be applied to multiple fields, and four of these applications are shown below:

Logistics and manufacturing: The Digital Twin can be applied to help the logistics system in a manufacturing setting, facilitating the planning of the logistics system, which allows visualization of how manufacturing will work in a particular scheme of planning the logistics system of manufacturing, with little modeling costs, while also using the organization DTs of organizational processes models that allow for real-time monitoring, manipulation of process parameters and optimization, recurring to the Industrial IoT and Artificial Intelligence (AI). Adapting the product to market requirements, making the detection of inefficiencies and reducing the time to introduce new products [18], [25]. This is possible through the flow of relevant information between DTs of individual products and components with more abstract DTs of production and logistic systems [26].

Kharchenko et al. [18] presents Digital Twin for a logistics system with the objective of determining the optimal location of manufacturing facilities to maximize productivity, allowing for real-time monitoring and to see how manufacturing will work in a particular scheme of planning the logistics system of manufacturing, with reduced modeling costs.

In the DT, data from processes, personal and equipment is collected using Industrial IoT, and then using an AI dedicated to decision support and a Digital Twin of the manufacturing process, all this information is given to a Digital Twin of logistics system which can be manipulated by a visual interface where the position of machines, machine proprieties and routes for transportation of products can be changed.

Healthcare: In Healthcare having a virtual replica of a patient, can be very beneficial for its health, as by having important parameters such as heart rate, blood pressure, respiration rate and sugar levels fed into a simulation of a person's body, gives the ability to monitor and predict future trends of a patient's health.

Elayan et al. [27] developed a novel Electrocardiogram (ECG) classifier, able to diagnose heart disease and detect heart problems, using a machine learning algorithm trained by using real-time data from ECG rhythms from different patients by using sensor electrodes that monitor the current health status, predicts future trends recurring to patient's medical history, enabling the health professionals prescribe the best treatments, test them in a safe environment and track responses.

Smart Cities: Ruohomäki et al. [28] implemented a Digital Twin in an already existing 3D model of the city of Helsinki, Finland, to increase the knowledge of possibilities to increase energy efficiency and renewable energy production especially in a renovation stage of the city to help to reach Helsinki's ambitious climate goals.

This was done by creating an open access energy data from sensors in building automation systems, where building owners can compare their energy consumption with the level of renovation needed in their respective homes to see potential energy improvements. These energy improvements can be in form of thermal isolation, solar panels and more energy efficient equipment [28].

Laboratory Scheduling: Lopes et al. [29] developed a digital twin of the quality control laboratory, to assist managers in the tasks of resource planning and scheduling. The authors implemented a discrete-event simulation model, said model was used as a testing platform to benchmark alternative governance models, scheduling heuristics and resource allocation policies intended to be deployed.

1.2.2 Dynamic Scheduling

Scheduling in this context has the goal of assigning a set of jobs, each having a set of operations that need to be scheduled in machines with the goal of reducing the total time to process all the jobs (makespan) and increase machine utilization.

Research in Scheduling started with Static Scheduling, that assumes that machines are always available and job attributes such as the processing time, release date and due date are fixed, but in the real world, machines can breakdown, orders can be late, operators might be unavailable, new urgent orders might arrive, there can be variations in processing time, especially in flexible systems, making the scheduling plan obsolete very quickly [30].

Therefore, changes must be made to the system in order to adapt to the actual conditions, to address this issue, in 1957, Jackson [31] defined the term Dynamic Scheduling. The three most common approaches of Dynamic Scheduling are Completely reactive scheduling, Robust pro-active scheduling and Predictive-reactive scheduling, which are explained bellow [32]:

Completely reactive scheduling: In this scheduling method, no firm schedule is made in advance and all decisions are made in real-time, the decisions are made using a dispatching rule to select the next job with the highest priority from a set of available jobs waiting to be processed by the next respective machine or resource that is free [32]. The priority of a job is determined based on job and machine attributes. Dispatching rules are quick, usually intuitive, and easy to implement (figure 1.4) [33].

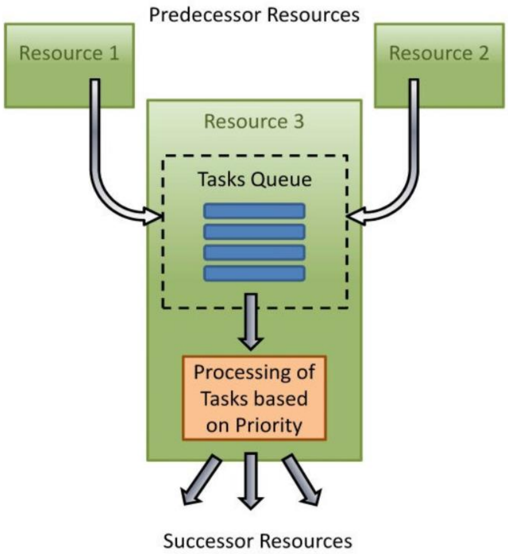

Figure 1. 4 Priority decision from a queue of tasks [32]

Robust pro-active scheduling: The concept is this approach revolves around creating predictive schedules by studying the main causes of disruptive events, the disruptions are measured based on actual completion measures compared to the originally planned completions, then the mitigation of these disruption are mitigated through simple adjustment to the activities durations [32].

Predictive-reactive scheduling: Is the most common of the three, the main idea is to build an initial baseline schedule, with the objective of optimizing the shop performance without considering disruptions, and afterwards this schedule is modified during execution responding to real-time events [34].

There are different policies in rescheduling, that answer to the question of when to reschedule, the three are:

- Periodic rescheduling: Rescheduling occurs between predefined time intervals
- Event-driven rescheduling policy: When the rescheduling is triggered by a disruptive real-time event
- Hybrid rescheduling policy (Rolling time horizon): This scheduling policy occurs both periodically and due to predefined events, that trigger a new rescheduling process.

The choice of the dispatching rules is important for dynamic scheduling, the most common ones are: [35] [36]

- First In First Out (FIFO)
- Last In Fist Out (LIFO)
- Shortest Processing Time (SPT)
- Longest Processing Time (LPT)
- Most Work Remaining (MWRK)
- Least Work Remaining (LWKR)
- Total Work (TWORK)

1.3 Robot System for Solution Preparation

The Cyber-physical space in this thesis, consists of the real asset and its DT, for the real asset, an already existing robot system for solution preparation, that can be applied in the chemical, food and pharmaceutical industries is used as a case study.

The prototype's purpose is to create liquid preparations of products in bottles, which is done by having an anthropomorphic robot with 8 different workstations with unique functions such as mixing, labelling, and stirring within the robot's range, with the robot being the resource responsible for the movement of bottles (figure 1.5).

Figure 1. 5: Robot System for Solution Preparation carrying a bottle

The already existing system processes a single bottle at a time, it's expected to reduce the makespan by recurring to parallel task scheduling, which means, having multiple entities being processed at a time, combined with testing different resource allocations and the right scheduling algorithm, to create a system capable of efficiently handling multiple bottles simultaneously and different costumer requests while allowing stakeholders to manipulate process and scheduling parameters and make real-time decisions, creating the desired flexible environment with Industry 5.0 ideals.

This thesis focuses on the simulation part of the DT, which is part of the environment comprised of Real System and Stakeholders and DT, shown in figure 1.6:

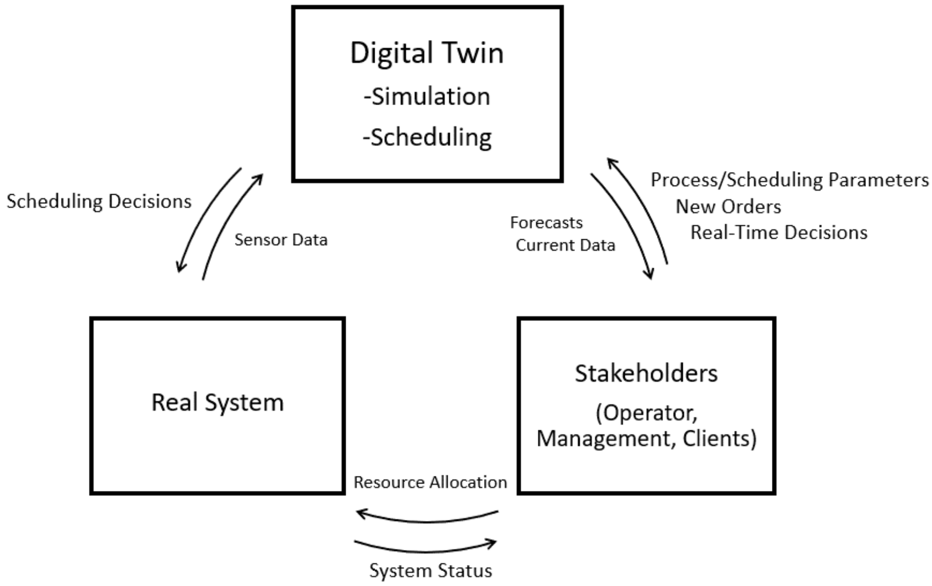

Figure 1. 6: Diagram of Real System, Stakeholders and DT

The Digital Twin receives both historical and real-time data, associated with the process flow of the manufacturing system, that is, processing times, transportation times and decisions that relate to the scheduling process, and with it, makes scheduling decisions, based on a chosen algorithm.

It works as a tool to help Stakeholders monitor the asset and do simulation runs to better understand the system and decide scheduling parameters, where resources should be allocated and to make real-time scheduling decisions with the constant flow of information from its real counterpart. It does not involve the modeling of the physical components, and their mechanical interaction or maintenance needs.

1.4 Objectives

To develop a Digital Twin capable of handling different customer requests and real-time decisions from operators, the objectives can be divided into 5 stages:

- Develop job workflows and decision-making processes to increase productivity of flexible production system.
- Create Digital Twin, that accurately emulates processes, workflows and decisions.

- Collect and process information related with the robot and the machines, to be an input to the Digital Twin, in order to make forecasts, and improve the system.
- Validate the system, recurring to actual data.
- Improve the system and maximize the equipment productivity by experimenting different resource allocations, algorithms and parameters.

1.5 Contributions

There is a lack of research containing mass customization and real-time changes from workers, as detailed in section 2.3, both involved in the topic of industry 5.0. This paper provides a methodology to analyze the behavior of flexible cyber-physical production systems. More specifically, a Digital Twin focused on simulation of a manufacturing system, which is capable of handling mass customization needs, and real-time changes from stakeholders was developed in Anylogic.

This Digital Twin is intended to make real-time scheduling decisions and forecasting the system's behavior for desired inputs and parameters such as client requests, heuristics, and resource configurations, to help managers better understand where the system can be improved.

The objectives in the previous section were achieved, a Digital Twin was developed, with the decision-making process recurring to parallel tasks, the simulation study increased equipment productivity, and decreased the makespan by 82%, making this concept fit to be applied in the future to the real asset, as well as establish a methodology to develop other Digital Twins of flexible manufacturing systems that can collaborate with workers.

1.6 Thesis outline

In Chapter 2, the concept of simulation and its types are described, in the end of the chapter, related work is exposed. Chapter 3 goes in depth into understanding the system, the development of the Digital Twin and the data collection. Chapter 4 presents on its first part, the calibration and validation of the model, and then, the results and analysis from the simulation study, in Chapter 5, the conclusions and future work are exposed.

Chapter 2

Discrete Event Systems Modelling

Computer simulation is a powerful tool used to analyze the performance of existing or newly designed systems by making use of mathematical or numerical techniques. This works by creating a model of the conceptual framework of the system, which then can be used for a wide variety of experiments with the system, and by analyzing the results, conclusions can be taken in order to help with decision making processes of the stakeholders [37], [38].

Analytical methods use mathematical reasoning to “solve” the model, for example using differential calculus to determine the minimum cost of a function, simulation instead uses computational techniques to “solve” mathematical models, therefore models are “run” instead of “solved”. [39] One type of simulation is discrete event modelling, where changes in the system occur at discrete times, these changes then affect the system depending on the chosen Time Advance Mechanism (TAM).

2.1 Time Advance Mechanisms

Simulation models are getting more complex, which increases the time to execute simulations as well as the necessary space, as the demand to have high fidelity to real models is increasing, making the selection of the appropriate Time Advance Mechanism (TAM) important, the simulation TAM is a method used to keep track and advance the evolution of time in a simulation, which is a variable commonly called “Simulation Clock” [37], [38], [40].

The two most common TAMs currently in use are the “next event” method, implemented in Discrete event Simulations (DES) and the “time step” or “fixed increment” method implemented in Discrete Time Simulations (DTS).

2.1.1 Discrete Time Approach

This method may have emerged from the need to study natural phenomena such as the law of gravity and mechanics which have continuous variables such as velocity and acceleration of rigid bodies or flow rate of fluids to represent the system. Real-time is continuous but a digital computer would take an infinite amount of time to represent it, so instead a “Simulation Clock” is introduced in which the time is discretized in order to be finite, this works by having constant time increments of Δt (that can be a second, an hour and so on), effectively “skipping time” between increments, this way, continuous time can be simulated on digital computers [38].

After each increment, a check is made on the whole system to determine if there was a state transition (st), during the previous Δt interval, and if so, the system updates the state variables, just as figure 2.1 illustrates, even though state transitions occur in continuous time, they are only considered at the end of the time intervals Δt.

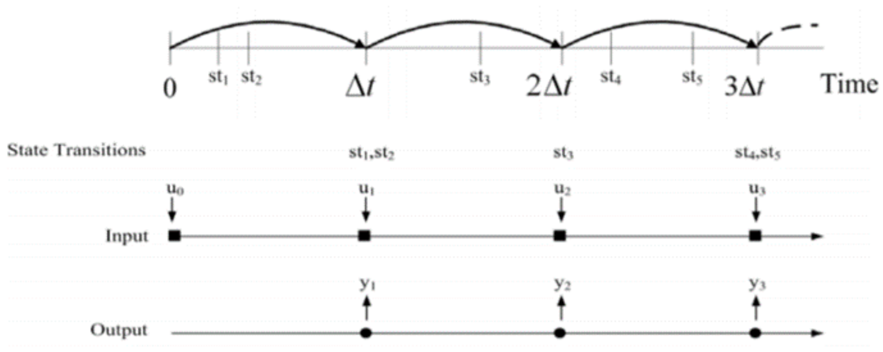

Figure 2. 1: DTS update intervals, with st_i meaning state transition i [38]

To increase accuracy, a smaller Δt would beneficial, but at the expense of longer execution time of the simulation [38]. Unfortunately, the selection of the size of the time step may not be that easy as for stiff systems, that is in slow moving components, as small-time steps might round off error over a large amount of integration steps, reducing the accuracy of the simulation.

2.1.2 Discrete Event Approach

Contrary to DTS, in Discrete Event Simulations (DES), the state transitions are driven by the combination of asynchronous and concurrent events, this way, events are the cause of change in the system, so the concept of time is not the driving force of the simulation and is instead a variable dependent on the state transitions, so the simulation effectively skips time between these, as figure 2.2 illustrates. The size of the time intervals can be deterministic or random, depending on the nature of the system to be modeled [38].

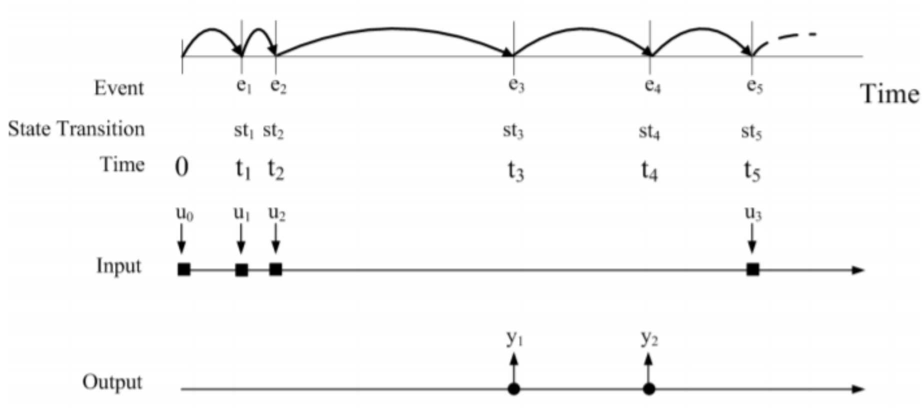

Figure 2. 2: DES update intervals, with st_i meaning state transition i [38]

This approach starts by setting the Simulation Clock to zero and a Future Event List (FEL) is created, in which all the times in which future events occur are calculated and stored in a list. Then the Simulation Clock moves towards the first event on the FEL, then the system updates its state variables associated with the current state transition, the FEL is also updated which may change the next state transition, afterwards, the Simulation Clock skips time to the next event on the FEL, this cycle is repeated until a chosen precondition finishes the simulation [39]. The figure 2.2 shows the flow of time relative to the events, and how the updates of the system, might bring new information (inputs) to the FEL or cause changes in the system (outputs)

2.2 Simulation Steps

To do a simulation study there is a set of steps that can be employed, Banks et al. [39] provides the steps bellow as guidelines, as well as figure 2.3, its notation is explained in chapter 3 in the BPMN section:

- Problem formulation: The problem described must be clearly understood, on some occasions it must be reformulated as the study progresses as more information might arrive.
- Setting of objectives and project plan: The objectives indicate the questions to be answered by simulation, necessary resources, and the expected results.
- Model conceptualization: Construction of the model, the model complexity shouldn't be larger than necessary to accomplish objectives, only the essence of the system is necessary.
- Data collection: Goes hand in hand with model building, later being necessary for model validation.
- Model translation: Enter the model into an adequate simulation software, in many instances reducing the amount of actual coding.
- Verification: Software must be prepared for the simulation model, therefore, parameters and logical structure must be correctly represented in the software, to accomplish this, it usually involves debugging.
- Validation: Measurement of discrepancies between the simulation model and the real counterpart, this process should be repeated until the model accuracy is acceptable.
- Experimental design: Choice of simulation alternative parameters, such as length of simulation runs and number of replications.

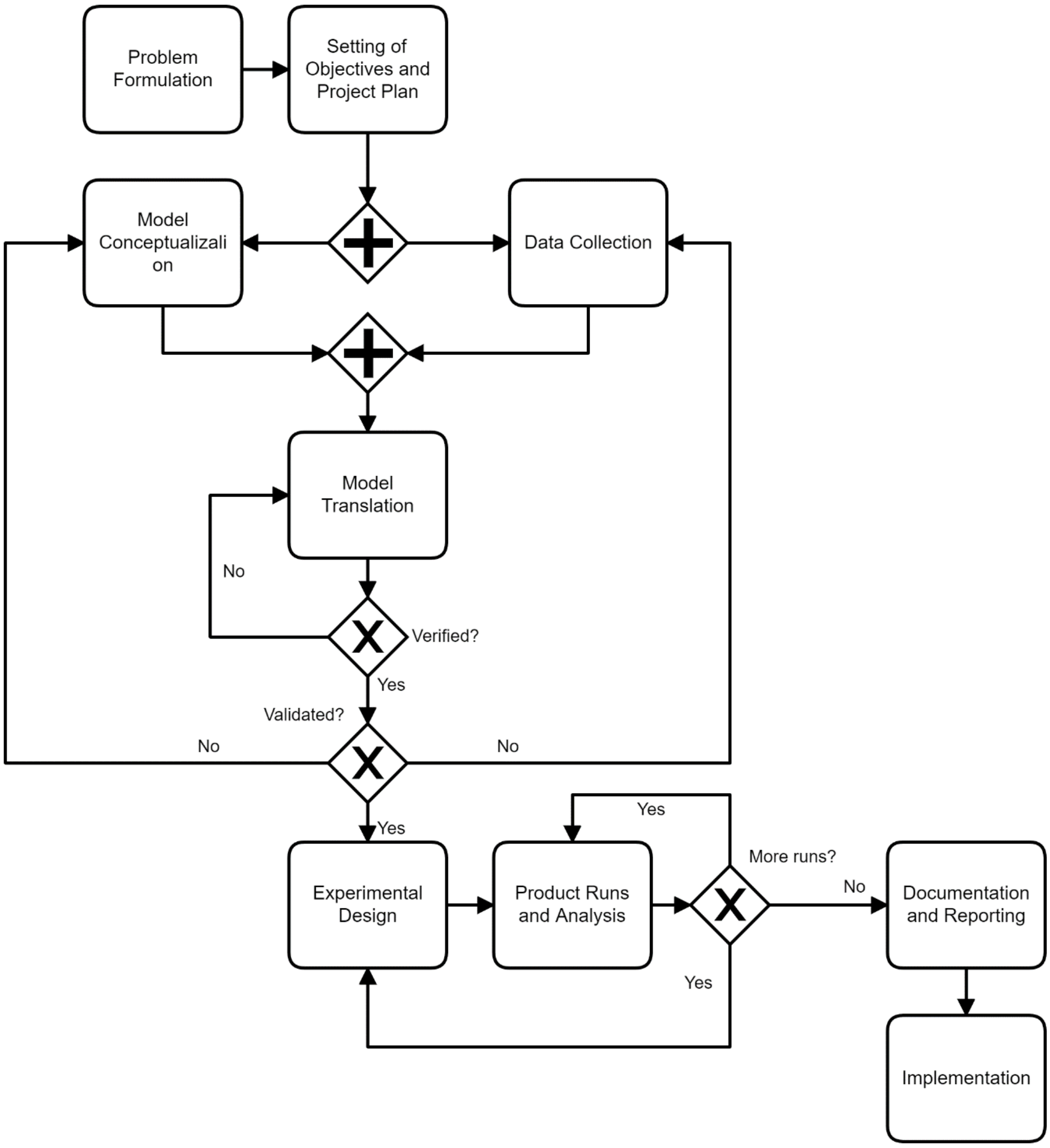

Figure 2. 3: BPMN of simulation steps, based on [39]

- Production runs and analysis: Used to estimate measures of performance for the system designs being simulated.
- Documentation and reporting: Program documentation is necessary if it is going to be used again or modified by different analysts, which is greatly facilitated by having the appropriate documentation. Progress reports give a chronology of the work done, which can give great insight to keep the project on the right course. All the results of the analysis should be in a final report, in which alternatives were compared.
- Implementation: If the user was thoroughly involved in the model building and understands the dynamics of the model, the more likely is the implementation to be successful.

2.3 Related Work

Yu et al. [41] Fused the concept of Digital Twin with job shop scheduling of a flexible job shop of a manufacturing workshop for box parts of an enterprise by recurring to a genetic algorithm. They used an integrated management scheduling model algorithm of intelligent IoT and cloud computing technologies to process production data. The authors created a Scheduling cloud platform that takes input from the sensors from the physical workshop which fuels data to its respective Digital Twin and also originates a fault prediction and diagnosis curve, the Digital Twin then gives to the Scheduling cloud platform simulation data and energy consumption diagrams to help planning the process steps. Yu et al. found advantageous to have access to a lot of data from the whole manufacturing process, helping to monitor the whole life cycle of the products, reduce energy consumption and predict failures in the processes.

Wladimir Hofmann et al. [42] presents a Digital Twin using a Python package called SimPy with real-time decision support for port operations, to deal with the issue of truck congestion in arrival gates, this is a flow shop problem with intermediate storage where the driving process of a truck is considered as parallel machines. The Digital Twin assists the dispatching operator in the decision-making process of releasing trucks whenever the port terminal is free, with the goal of reducing the probability of deadline violations due to low utilization of bottleneck resources.

The Digital Twin receives both present (from sensors) and past information, using IoT, from the registered trucks, and by using a dispatching algorithm based on starvation avoidance, the Digital Twin then enables different dispatching policies, that are defined by different safety buffer levels (processes with a high safety parameter are more likely to have early arrivals), to be evaluated and presented to the dispatching operator, he can then discuss with the involved stakeholders the best combination of safety parameters.

Karagiannis et al. [43] addressed the issue of how hard automation solutions that increase productivity, end up not allowing industries to adapt to market changes and system malfunction, by utilizing an existing consumer goods industrial production line, responsible for the assembly of shavers as a case study. The authors developed a DES model, in Witness Horizon Simulation tool to include industrial robots with smart mechatronic devices and smart algorithms in the existing system, to better manipulate small components, that allow for the accommodation of multiple products in the same line.

The final simulation model offers the possibility to test all the probable occurrences in the assembly line, by the manipulation of parameters, such as machine breakdowns or introduction of new parts in a risk-free virtual environment. With the simulation results Karagiannis et al. presented the main advantages of the inclusion of the mentioned technologies to be:

- Increased flexibility without the need to redesign a production cell.
- Rise in reliability in terms of adapting to unsuspected breakdowns.

To the authors knowledge, there is not a lot of research the topic that bring together the concept of Digital Twin working with a human counterpart in a flexible environment, the literature mostly present somewhat stochastic conditions, machine breakdowns, new orders, late arrivals and workers unavailable as proprieties that make the system dynamic, but rarely the inclusion of a human counterpart, capable of changing processing times and workflow, also, the flexibility of the manufacturing systems, tends to come from supply chain flexibility, not from the manufacturing process itself [32], [34].

Chapter 3

Developed Digital Twin

The first step to make the necessary changes to adapt this prototype to the necessities presented earlier, is to know the workings of the already existing system which will also help with the validation of the Digital Twin.

The Digital Twin gives the ability to better understand the Stakeholder/Real Asset/Digital Twin environment, that is, how real-time changes either caused by the real asset or decisions by stakeholders affect the DT, and how the Digital Twin responds to these changes, which will help to recognize where the system can be improved.

3.1 Model Conceptualization

In this system, a bottle is considered as the entity that goes through the necessary processes, it first starts in an Entrance Storage (ES), then it's transported by the robot manipulator through each workstation to be processed, until it ends in the Final Storage (FS), both the ES and the FS use a rotating storing device, the workstations (Ws) and their positions relative to the robot are displayed and explained bellow (figure 3.1):

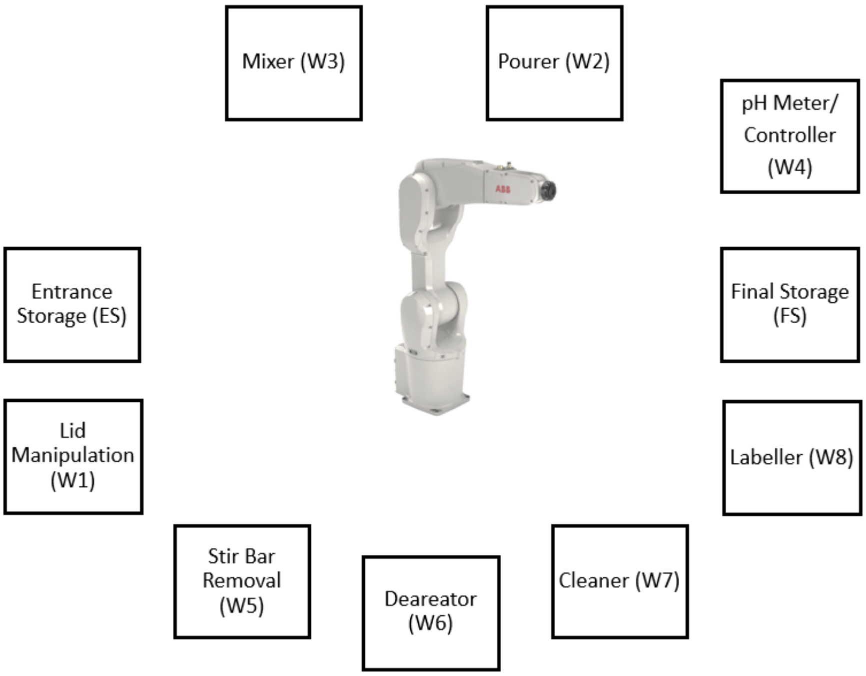

Figure 3. 1: Position of Workstations and Robot of the prototype

Workstation 1 (W1): This workstation specializes in the manipulation of the bottle lid, it can unscrew, loosen, and tighten the lid, so each entity has to go through W1 multiple times.

Workstation 2 (W2): Has the function of pouring reagents in the bottle, it can have up to three different reagents, and it has the ability to freely mix different reagent and choose the necessary quantities automatically, the time that it takes to finish the process is related with quantity of liquid and the number of reagents.

Workstation 3 (W3): A magnetic stirrer that mixes the liquid inside the bottle, it works by dropping a stir bar into the solution, and then, by using a set of stationary electromagnets or a rotating magnet, placed beneath the bottle with the liquid, the stir bar spins very quickly, therefore mixing the reagents, just like with the workstation 2, the process time depends on the quantity of liquid and number of reagents.

Workstation 4 (W4): pH Meter and Controller, this workstation has the goal of making sure that the pH of the solution is within the client's expectations, and if not, it corrects the pH of the solution to the necessary amount and the bottle is sent back to W3 for mixing and back to W4 to recheck the pH, also, depending on the judgement of a worker and the necessities of the client, this process might be skipped. Even though it's not in the existing system, it's going to be included in the future, so it's not employed in the validation procedure with the existing system.

Workstation 5 (W5): Has the purpose of removing the stir bar that was placed inside the bottle in the W3.

Workstation 6 (W6): Deaerator, depending on the client's request, the mixing and stirring the of the liquid solution, may create bubbles at the solid-liquid interface as well as leaving the solution with undesired dissolved gasses such as oxygen, the Circulating Water Bath applies temperature evenly in the solution helping with this problem. The time on the W6 is different depending on client requests, and a worker change the time on this Workstation in real-time, according to his judgement.

Workstation 7 (W7): Cleaner, this device removes the water droplets, from the W6, that are still on the bottle outside surface due to surface tension, it's a cylinder shaped device in a vertical position with multiple hoses with flowing hot hair towards the inside of this cylinder, when the robot arrives grappling a bottle, it does a slow descending and upward motion with the bottle inside the cylinder, leaving when the bottle is dry enough.

Workstation 8 (W8): Labelling Workstation, as the name suggests, it places a tag on the bottle, it has an embedded tag printer and an automated robot places that tag, then by using a sponge, this tag is firmly attached to the bottle, this step is optional.

Each movement of the robot when it is carrying a bottle is also considered as a process $M_{ii'}$, the first one M_{01} describing the movement from ES to W1, M_{12} from W1 to W2 and so on, in total there are 13 movements, and each bottle has its own order of movement, and may repeat or skip some, depending on the workflow, these are shown in figure 3.7.

As its intended to create a model with parallel tasks, every time the robot finishes placing a bottle in a workstation, a chosen dispatching rule determines what bottle should the robot attend to next, and so, the robot moves without carrying a container when changing between tasks, this motion time is defined as pre movement time Mp_{pi}.

3.1.1 Graphical Representation of Workflow

To better understand how the processes interact with each other, this can be achieved by using Business Process Model and Notation (BPMN), a visual modeling language for defining enterprise process workflows. It was first created in 2004, with the objective to provide a standardized notation that is intuitive and easy to understand by all stakeholders, such as, business users, business analysts, software developers, and data architects [44]. Its popularity kept rising since its creation, a study revealed that in 2013 [45], 64% of companies were interested in adopting the BPMN standard, the latest version (BPMN 2.0.2) was published by ISO as the 2013 edition standard: ISO/IEC 19510 [44].

This way, relevant stakeholders can respond to any issues identified in the process more effectively, BPMN provides a simple, yet rich notations that are easily understood by both technical and non-technical business stake holders, often bridging the communication gap between business process design and implementation. With its goal being the usage by technical experts responsible for process implementation, business analysts who create and improve the processes and managers who monitor and control the processes [44].

In BPMN the processes are represented by using diagrams with a series of graphical elements, grouped by different categories for easy understanding. There are four basic categories of BPMN elements, each representing a unique aspect of business processes.

3.1.1.1 BPMN

Swimlanes

Swimlanes represent participants in a business process, processes that are inside swimlanes are associated with the respective participant, examples of participants are costumer, account department, factory floor and development team.

There are two types of swim lanes, pools and lanes, pools represent participants which can be departments or roles of individuals for example, and lanes are a sub partition of a pool, for example, if you have a pool called factory, the lanes can be production and quality control, as they represent subsections of the pool (figure 3.2).

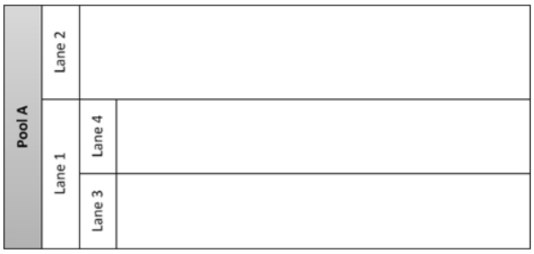

Figure 3. 2: Representation of pools and lanes in BPMN [46]

Activities

Activities are works done within a business process represented by a rectangle, there are two types, Task and Sub-Process, the first is used when it is not possible, or it doesn't make sense to break down a work, and the second is used when its necessary to represent multiple works within one rectangle (figure 3.3).

Figure 3. 3: Representation of tasks and subprocesses in BPMN

Events

Events represent something that happens and impacts the business process, there are 3 types Start Event, Intermediate Event and End Event, these can represent for example, an email confirmation, a part breakdown, an arriving order, or for the End Event, the end of a process.

Figure 3. 4: Representation of events in BPMN [46]

Gateways

These control the workflow of the BPMN, they represent between different paths, there are 6 types:

- Exclusive Gateway, controls the flow by making an exclusive choice, only allowing one flow to be traversed.

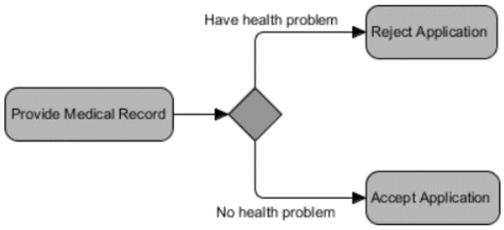

Figure 3. 5: Representation of exclusive gateway in BPMN

- Inclusive Gateway, all the conditions of outgoing flow are evaluated. All flows with positive result will be traversed.
- Parallel Gateway, it has parallel flows that have to be executed at the same time.
- Event-Based Gateway, diagonal shaped and are used when parallel flows depend on events, for example when waiting for a confirmation, can be a yes, no and if those aren't triggered, another event happens.
- Parallel Event-Based, combination of Event-Based Gateway and Parallel Gateway, although it will allow multiple events to pass through and start the corresponding portion of the process, it does not wait for all of the events to arrive.
- Complex Decision Gateway, allows for a more expressive decision within a business process. Multiple factors, rules, and analyses can all combine to yield results. The analysis should result in at least one path always being taken.

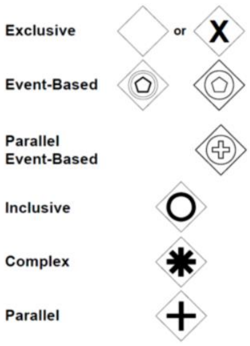

Figure 3. 6: Representation of gateways in BPMN

3.1.1.2 Graphical Representation

With the knowledge it is possible to show the BPMN of the workflow of each Job or bottle, as shown in figure 3.7, it does not represent the movements as processes, as they are only written as comments, and when the robot holds the bottle, it doesn't show the interaction with the worker.

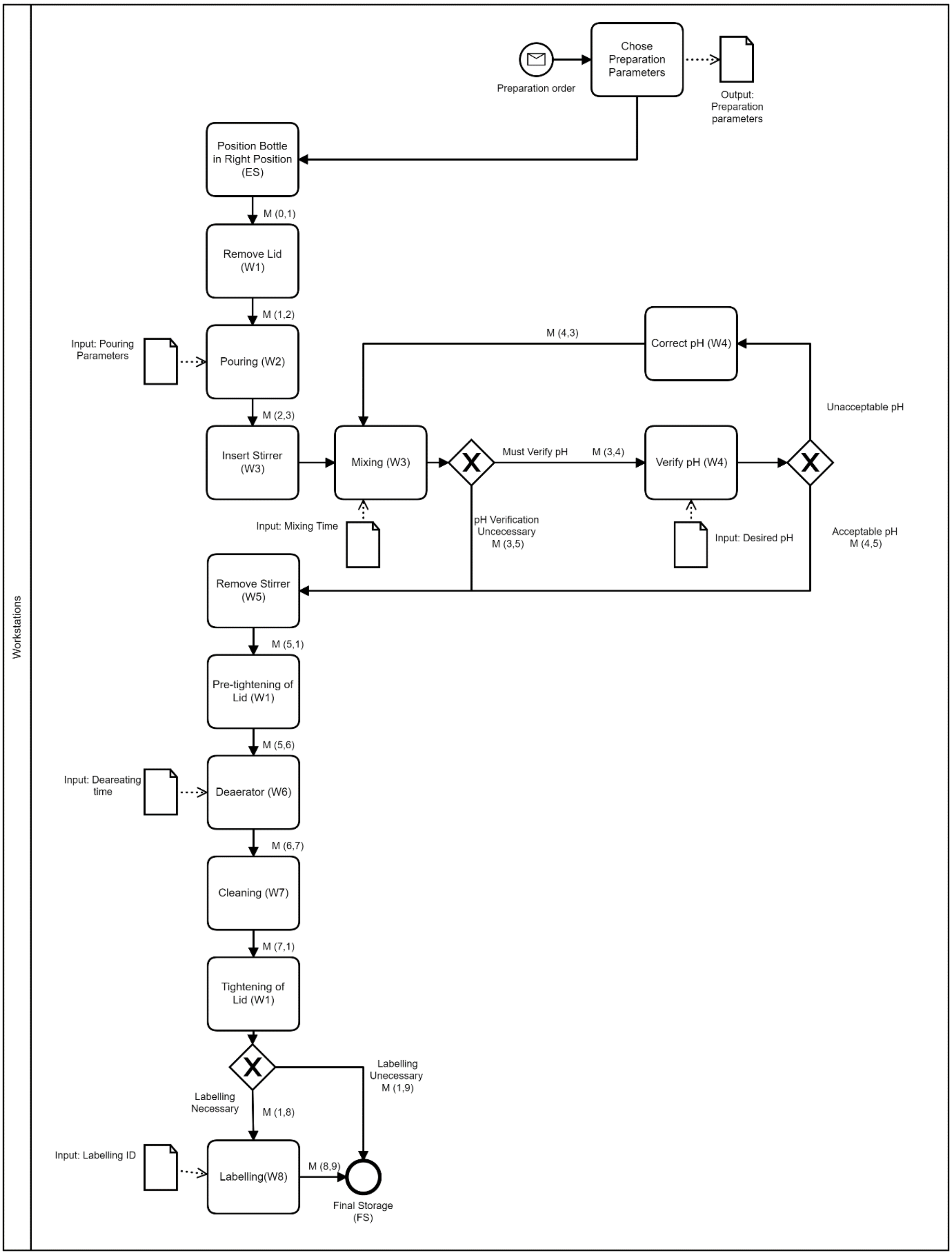

Figure 3. 7: Workflow of each client request through the workstations in BPMN, the movements are represented only by the M letter

3.1.2 Scheduling and Algorithm

The industrial prototype makes liquid solutions in bottles, which can be described as jobs $J_1, J_2, \dots, J_n$ to be scheduled in the machines W1 to W8 with the order represented in figure 3.7, where each job has a specific route through the machines depending on customer requests and real-time decisions.

Simulation was utilized to determine the system's performance, when identical parallel machines of chosen Ws, are added, they are added if they disrupt the workflow and increase the makespan, by either having a long processing time, or being a bottleneck.

The processing operation of job J_j is defined as an operation O_{ij} with a processing time t_{ij}, each job contains a set of operations $O_{1j}, O_{2j}, \dots, O_{ij}$, where operation O_{ij} can only start after $O_{i-1\,j}$ finishes. Because the robot needs to change its position between jobs (pre movement), there is an extra movement between each operation that is between O_{ij} and $O_{i'j'}$ there is a $M_{ii'}$ movement time.

The dynamic part comes from the fact that there are real-time events and decisions, such as:

- Number of iterations between W3 and W4 until the pH is correct either automatically or by the workers decision.
- Real-time changes in the process time in W3 and W6.
- Preparation orders can arrive at any moment for processing.
- Stochasticity of processes.

These dynamic processes make the schedule building a difficult task, as it quickly gets obsolete, making rescheduling too frequent and ineffective, this makes a completely reactive scheduling an adequate option to apply in this case study.

To address this issue, traditional dispatching rules such as, the Shortest Processing Time (SPT), Longest Processing Time (LPT) and Least Work Remaining (LWR) were employed, which define the priority of jobs currently not being processed, the Job j with highest priority according to the chosen dispatching rule, is the next one to be processed. This procedure occurs according to Figure 3.8, and the priority is set to zero if availability conditions (section 3.1.4.1) are not met, the dispatching rules have the following equations:

Shortest Processing Time (SPT), each job has an associated priority P_j with j as the job number and t_{ij} as the processing time of operation i defined as:

$$P_j = \frac{1}{t_{ij} + \varepsilon} \tag{3.1}$$

If the next movement will be towards the Final Storage, t_{ij} will be zero and the priority will be infinite, to prevent this, a small number (ε) is utilized in the priority equation.

Longest Processing Time (LPT):

$$P_j = t_{ij} + \varepsilon \tag{3.2}$$

In the case of LPT, the small number (ε) is used to distinguish the priority from zero, as this means, that the job is not available (figure 3.8).

Least Work Remaining (LWR), sums all the processing times of a job j, from the current operation c, until the last one r:

$$\sum_{i=c}^{r}(t_{ij}) + \varepsilon \tag{3.3}$$

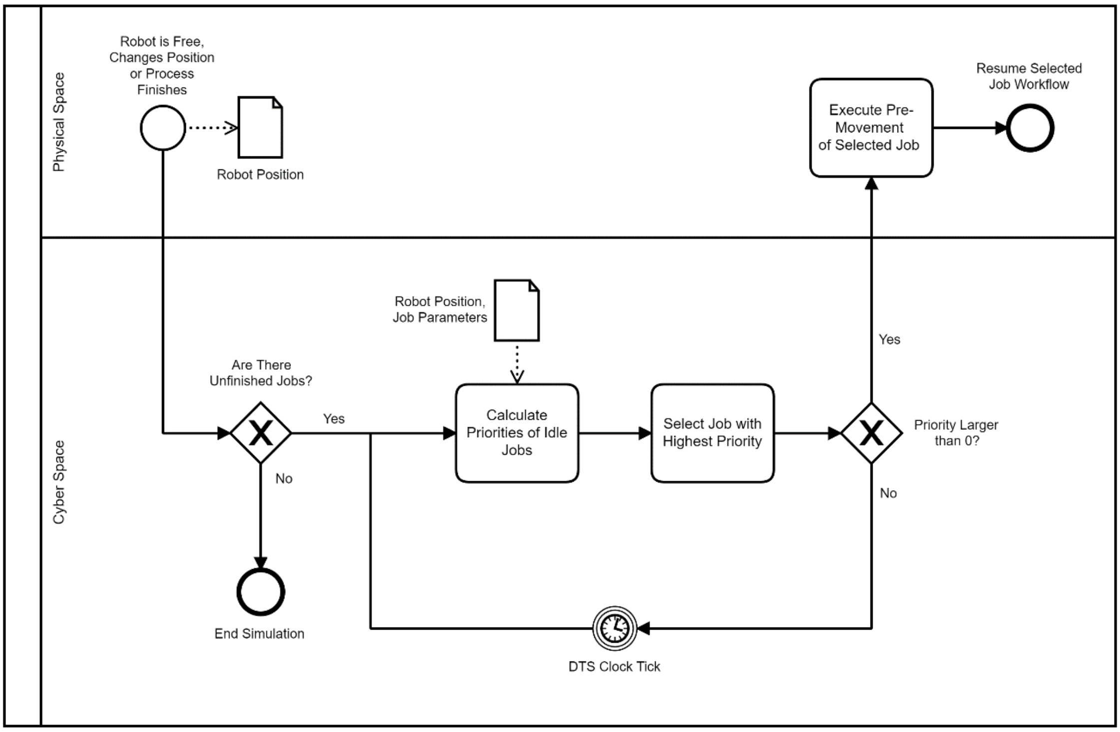

Figure 3. 8: Decision workflow for the SMT, SPT, LPT and LWR dispatching rules in BPMN

Since the transportation time is long, which can be seen in the chapter 3: Data Processing, the robot's movement can be relevant in scheduling decisions, two variation algorithms based on the movement time were employed, to analyze how they fare against traditional scheduling algorithms:

- Shortest Movement Time (SMT): which prioritizes Jobs that are closer to the robot's current position, based on the time it would take for the robot to reach the desired Ws, the priority of a job is defined by the equation 3.4, and the figure 3.8 represents the decision process.
- Current Shortest Movement Time (CSMT): When the robot places a bottle in a workstation, it might be beneficial to wait for the process to finish, and transport that same bottle, CSMT employs the SMT idea for jobs currently not being processed or free jobs, and it compares the

priority of these jobs (equation 3.4), with a priority of the job that the robot just transported, defined as the current priority (equation 3.5), which is based on the processing time, the decision making process can be seen in figure 3.9.

Free Jobs: the priority equation P_j with j as the job number, Mp_{pi} being the pre movement time from current location of the robot p to the workstation k associated with the job j shown below as well as the BPMN of the priority decision:

$$Pi_j = \frac{1}{Mp_{pi} + \varepsilon} \tag{3.4}$$

Current Job: Current job is defined as the job in which the robot is about to finish or just finished, transporting or processing, depending on the operation, its priority equation is defined as:

$$Pc_j = \frac{1}{t_{ij} + \varepsilon} \tag{3.5}$$

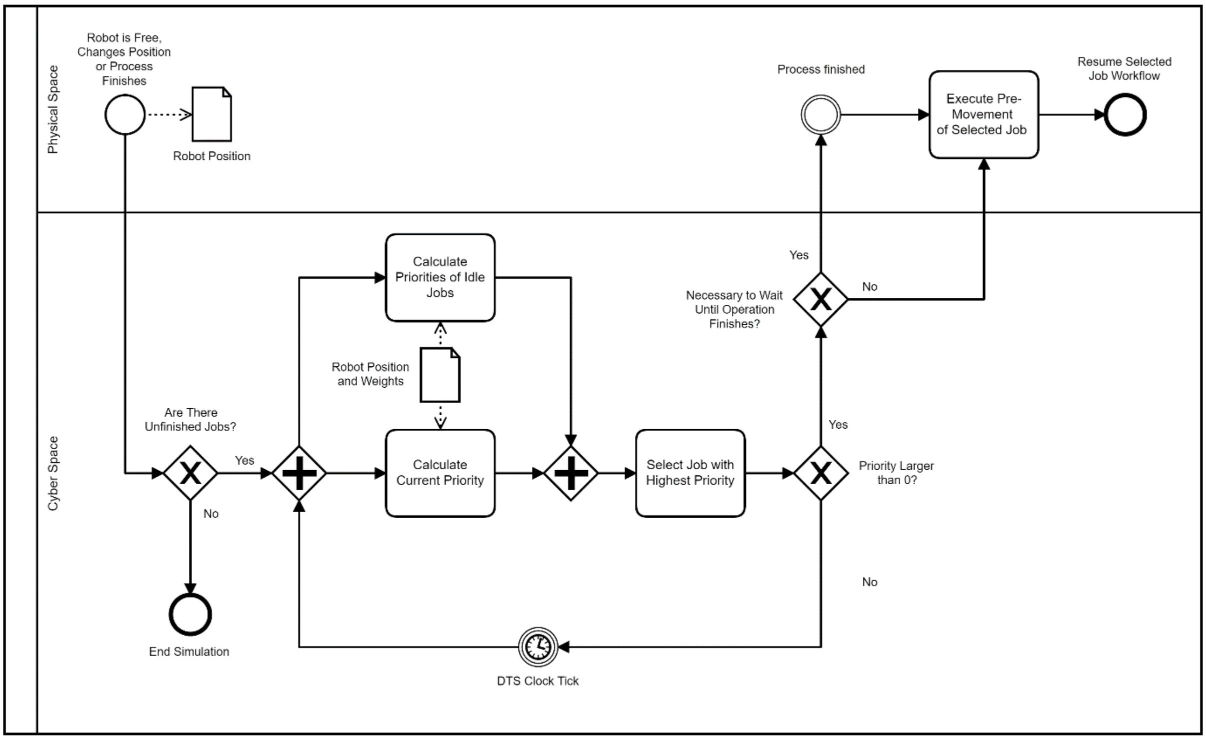

Figure 3. 9: Decision workflow for the CSMT dispatching rule

From a simulation point of view, the system is considered as a mixed event simulation, as updates in the system occur when the robot is free, then the time “jumps” to the next event, in which the robot is again free or a process ends. When the robot is free and the system cannot move forward because priorities are zero, due to availability conditions being violated, the simulation employs the DTS clock, that recalculates the priorities every set period of time, until the availability conditions allow the priority value to be larger than zero.

3.1.4.1 Availability

The priority equation is only applied if some conditions derived from Workstation availability are met, otherwise, the priority value is zero, these are:

- The next workstation of a job is not full, that is, all the machines are not seized.
- Moving a container to the next workstation won't stop the flow of tasks, for example, considering that each workstation only has one available machine, jobs J_1, J_2 and J_3 are currently being processed in workstations W2, W3 and W5 respectively, the next operations of these jobs are in W3, W5 and W1, then a new job J_4 arrives and the robot moves it from ES to W1, with its next operation being on W2, this movement originates a standstill in the whole system, because all the bottles in workstations cannot move anywhere, and it can only be solved by removing a bottle temporarily, this is better explained in Figure 3.10:

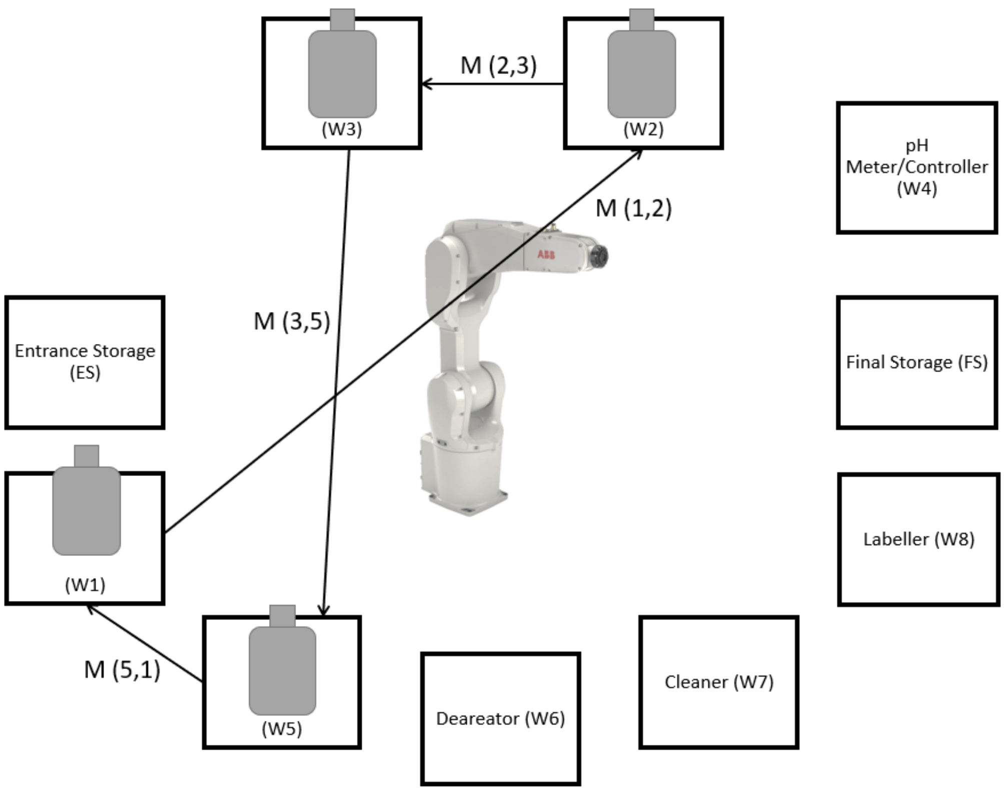

Figure 3. 10: Representation of a common standstill in the system, where placing a bottle in the W1 causes the workflow to stop

3.2 Model Translation

To develop the Digital Twin, the Anylogic software was utilized, this software includes a graphical modeling language that allows for discrete event, system dynamics (discrete time) and agent-based methodologies.

Anylogic's flexibility allows it to be used for a multitude of areas of application, some of which are: manufacturing, healthcare, supply chains, maintenance scheduling, defense systems, pedestrian and social dynamics and others, with the help of libraries that facilitate the modeling process such as the Process Modeling, Material handling, Pedestrian and Fluid Library [47].

As graphical modeling only allows for simple applications, Anylogic allows Java code to extend the model as things such as probability distributions, equations, test conditions containing properties of different objects, and algorithms need a textual scripting language.

3.2.1 Parameters

When a client requests a preparation order, all the relevant information needs to be described in parameters for the system to allow for the flow of processes, part of this information is fixed, part of it is updated throughout the simulation time, these parameters are either defined in a excel sheet or as a parameter for the entity in Anylogic. In some cases, the data type of the parameters might be of Boolean type, characterized by being either true or false, instead of a numerical format.

In Anylogic, each entity or bottle is defined as an agent with different parameters, such as pre-movement time, movement time, current or idle job priority, W3-W4 cycle, other parameters are:

- Row: The row is associated with a bottle on a list of preparation requests, basically j , the indicator of a job J_j, allows the model to access all relevant information of a request, such as processing times, if pH correction or a receipt is necessary and others.
- Column: Is the index i, the number of the operation, with the row and column it is possible to obtain the processing time of each operation from a list of jobs.
- Processing Time: From recipe, may be affected by an operator or the stochasticity of a process.
- Adjustment: Internal variable of the model, with value from a chosen distribution, defines how many times more the entity must go through W4, is reduced by one, every time the entity is processed by W4, when it reaches zero, the bottle goes to W5.
- Next: Defines the next workstation where the bottle needs to go.
- Bottle Position: Defines the WS in which the bottle is located.
- W3-W4 cycle: True if a bottle needs to check pH, and is either in W3 or W4.
- Pre-movement time: Movement of the robot when it is not carrying a bottle, determined based on bottle position and robot position, utilized when the robot is assigned for another task.
- Movement Time: Utilized when the robot is carrying a container, it's value depends on the location of the next operation and bottle position.
- Idle Job Priority: Stores the calculated priority of a free job.

- Current Job Priority: Stores the priority value of the job that the robot just released.

As well as general parameters such as:

- Robot Position: The WS where the robot is currently.
- Home: Describes if the bottle needs to go to home position.
- Number of Bottles: Defines the total number of jobs.

These parameters give the model relevant information that helps to make scheduling decisions and choose the correct workflow through processes.

3.2.2 Workstations Translation

The model is divided in workstations, each one with a set of Anylogic block such as Seize, Resource Pool, Move by Crane, Release and Delay.

These activities can be described in Anylogic recurring to the process modeling library and the material handling library, the robot is described in Anylogic as a jib-crane, as the software does not have an animated robot, it emulates the movements and interaction with the bottles well enough in a visualization setting, for this the block (MovebyCrane) was utilized, which allows the modeler to define the trajectory and transportation time of the entities.

To define the number of identical parallel machines of a workstation, Anylogic has the *resource pool* object in which the number of machines, where each machine is a resource, can be chosen, to select a machine from a set, the *seize* block takes one machine from the resource pool every time an entity flows through it, when moving a bottle to a new workstation, the workstation of the previous operation is freed, therefore, the resource that was seized in the previous operation is released by the *Release* block, as the bottle is no longer on that workstation, the process flow can be seen in figure 3.11 and an overview of the Anylogic model can be seen in figure 3.12.

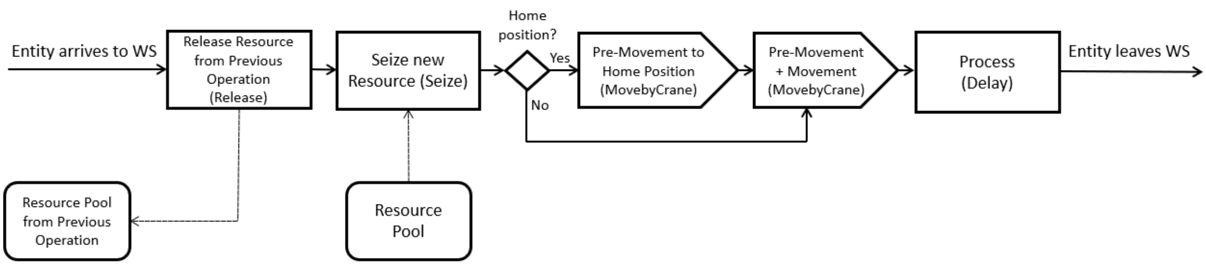

Figure 3. 11: Representation of workstations in the Anylogic model

When the entity arrives to the WS, the Robot and Bottle Position parameters are updated to the current WS, then at the entrance of the Seize block, that calls for the recalculation of Priorities, then at the end of the last MovebyCrane block, the next job with the highest priority is chosen, that is the Direct parameter is true, if an Idle Priority is larger, this is the model translation of the flowchart in figure 3.8, or 3.9 if the CSMT rule is in use.

After the bottle finishes the Delay, the Column parameter as well as the processing time is updated to the next operation, depending on the recipe and real-time parameters (Row and Adjustment), as well as the movement time and in the case of Direct being true the pre-movement is also calculated.

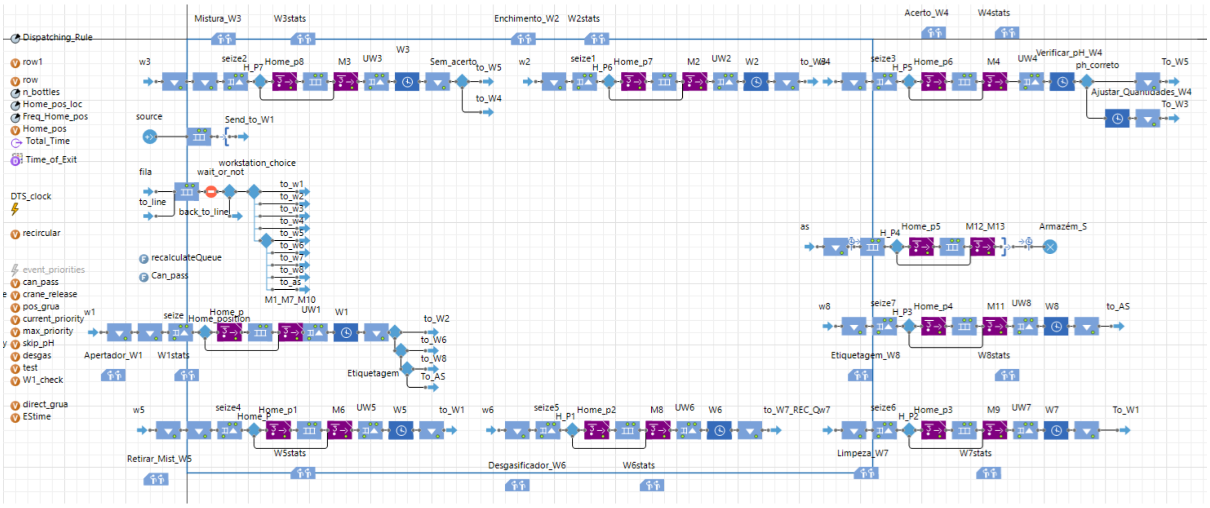

Figure 3. 12: Anylogic model overview

To simulate the DTS clock, that kicks in if the workflow is stopped, the Event component from the Agent library was put to use, this Event component has a written java code that checks if the robot is idle, and if so, it recalculates the idle priorities periodically over a chosen time period.

3.2.3 Visualization

Anylogic also supports both 2D and 3D animation, which allows the modeler to visualize and better understand the model, as well as to present the model in an accessible way to relevant stakeholders, which can be seen in figure 3.13.

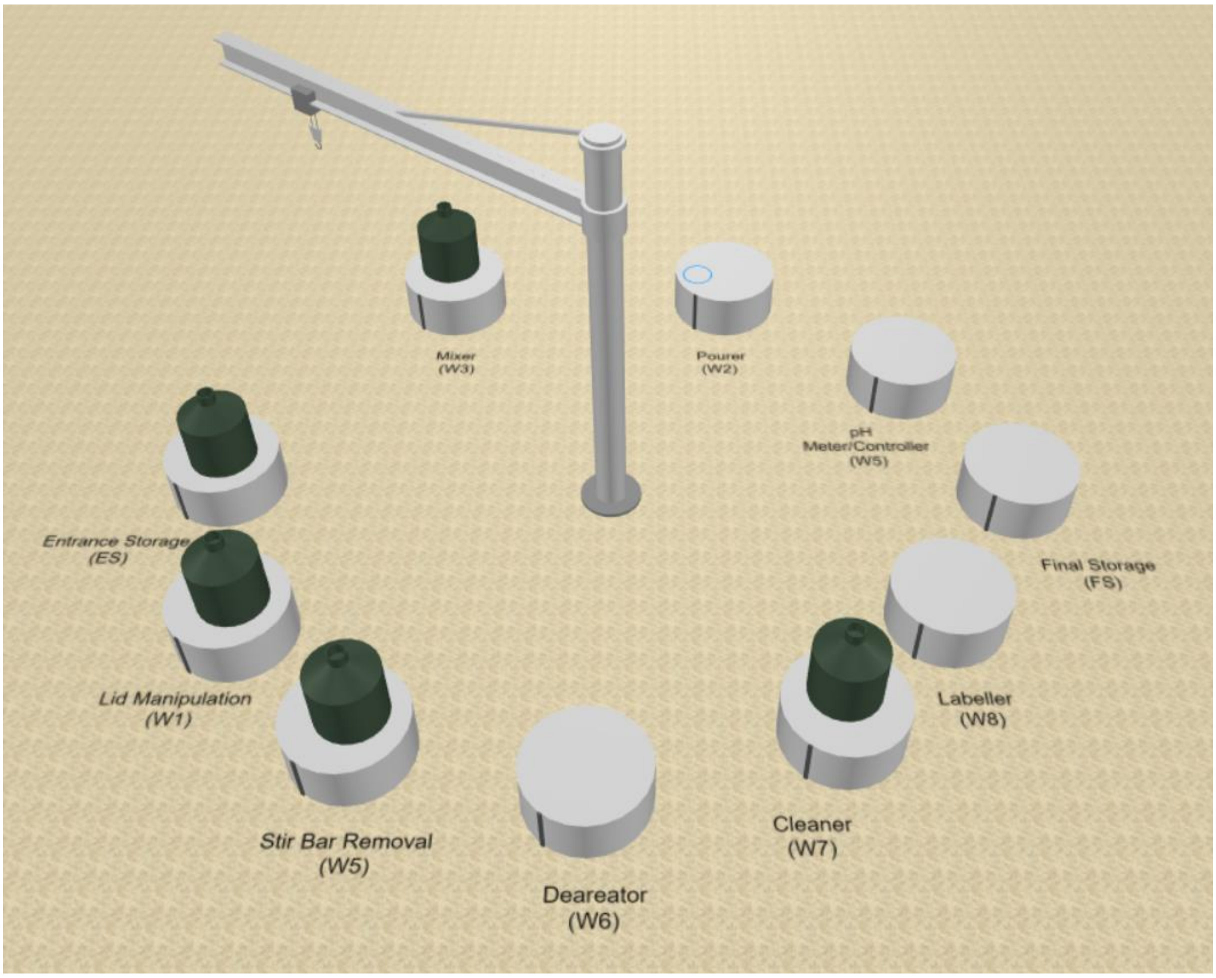

Figure 3. 13: Visualization window in Anylogic during a simulation run

The animation also helps the modeler in simulation or forecasting, as it provides another angle to understand the model, and make sure its functioning correctly, as visualizing the robot movement and interactions with the bottles is an easier way of understanding the model while its running, in figure 3.13, the 3D animation of the model, during a simulation run can be consulted.

3.3 Data Processing

One of the challenges of this project was the data collection since the case study is a prototype, which makes documents with relevant data (processing times, different orders) difficult to find, for this reason, a time study of the physical asset was conducted to obtain information related with the processing and movement of the robot, for information such as requests from the customers and real-time changes, experts in the field that were involved in the development of this equipment were consulted.

Time studies nowadays entirely focused on knowing how long a job takes, as well as getting fundamental information regarding how a process works [48]. Time information is obtained by direct and continuous observation of a task, using time keeping devices such as stop watches, and more recently, cameras to film the time necessary to finish a process, it is commonly used for repetitive or cyclic tasks and there is a great variety of work performed, the Industrial Engineering Terminology Standard defines times study as:

"A work measurement technique consisting of careful time measurement of the task with a time measuring instrument, adjusted for any observed variance from normal effort or pace and to allow adequate time for such items as foreign elements, unavoidable or machine delays, rest to overcome fatigue, and personal needs"[49].

According to Magagnotti et al. (2012) [50], a comprehensive time study consists of the following steps:

- Study goal: Is affected by what knowledge is necessary to acquire, the foreseen use of the results and the available resources.
- Experimental design: Process of planning the time experiment to ensure the right type of data with sufficient sample size is obtained.
- Measurements in the field: Describes how to record different types of data on different levels, such as: shift level, cycle level and element level which helps to better understand the process dynamics.
- Data analysis: Relates with experimental design as a specific one is geared towards a specific data analysis, in which the data is statistically described commonly by mean, standard deviation, minimum and maximum.

3.3.1 Robot Movement Study

To collect data related with the robot's movement time, it's necessary to understand its motion, since the already existing system only provides a limited set of movements, shown in figure 4.1, some extrapolations need to be done to obtain all the possible movements for a more flexible manufacturing environment.

The robot's model is ABB IRB 1200, from data available in the user's manual and product specifications [51], and the time study done on the robot, the following information was obtained:

- The time associated with the movements is fixed and is shown in table 3.1:

Mov.	$M_{0,1}$	$M_{1,2}$	$M_{2,3}$	$M_{3,5}$	$M_{3,4}$	$M_{4,5}$	$M_{4,3}$	$M_{5,1}$	$M_{1,6}$	$M_{6,7}$	$M_{7,1}$	$M_{1,8}$	$M_{8,9}$	$M_{1,9}$
Time (s)	24	14	10	12	12	25	12	37	29	5	24	18	42	40

Table 3. 1: Original movement times of the robot

- It's home position, which is a fixed location where all joint values are zero, its used for calibration both when the robot starts its working cycle and during it to reduce accumulated joint error, is located over W2, the angular location in radians (rad) of each workstation relative to the home position, in a counterclockwise rotation is shown in table 3.2:

	ES	W1	W2	W3	W4	W5	W6	W7	W8	ES	Home
rad.	1.9	2.4	0.0	0.5	5.8	2.7	3.4	4.1	4.5	5.0	0.0

Table 3. 2: Angular location in radians of the workstations, relative to the home position

- The robot has a working range of ±170º from the home position, and its dead spot. Which is the angle that the robot cannot rotate to, is between W5 and W6.
- Since it is not relevant the movement of all joints, and most of the movements necessary for the robot do not exist yet, so they need to be extrapolated. With this purpose in mind, the robot's movement can be simplified, through the division of a single movement in three components, the approach motion time (ma), the rotational motion time (mr) and the exit motion time (me), these are to approach the workstation and place the bottle, to do the rotation between workstations, and to take a bottle from a workstation respectively, the motion of retracting the robotic arm after the approach and the exit are included in these motion times.

  Each workstation has its own approach motion time (ma), and exit motion time (me), and every pair of workstations, have a common rotation motion.

  When switching between jobs, the pre movement time (Mp_{pi}) is employed, it consists of the rotation motion without a bottle, from its current position (p), to the workstation where the next movement starts (i), then the robot performs the exit motion (me) for the Ws (i), grabbing the bottle, and retracting it, this is shown in figure 3.14, and defined by equation 3.6b.

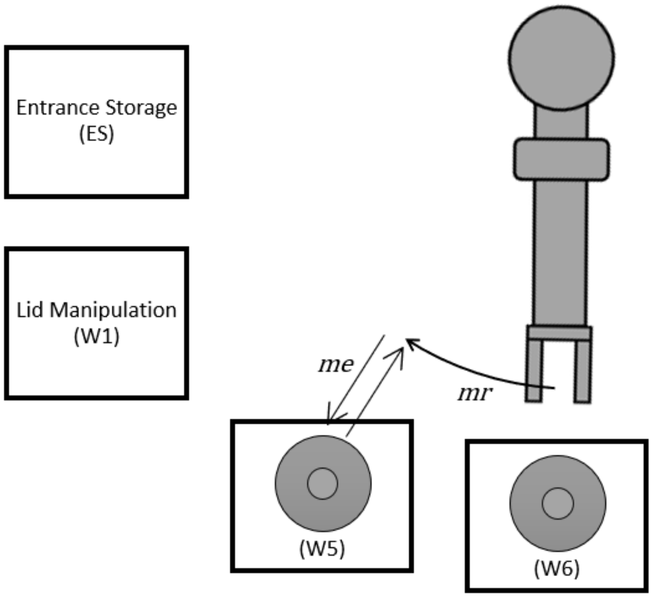

Figure 3. 14: Pre movement from robot's current position W6 to W5

The movement time ($M_{ii'}$) consists of the rotation motion time (mr) with a bottle, from the workstation where the movement starts (i), to the next workstation (i'), where the robot leaves the bottle and retracts, with the approach motion time (ma), this can be shown in figure 3.15 and is defined by equation 3.6a.

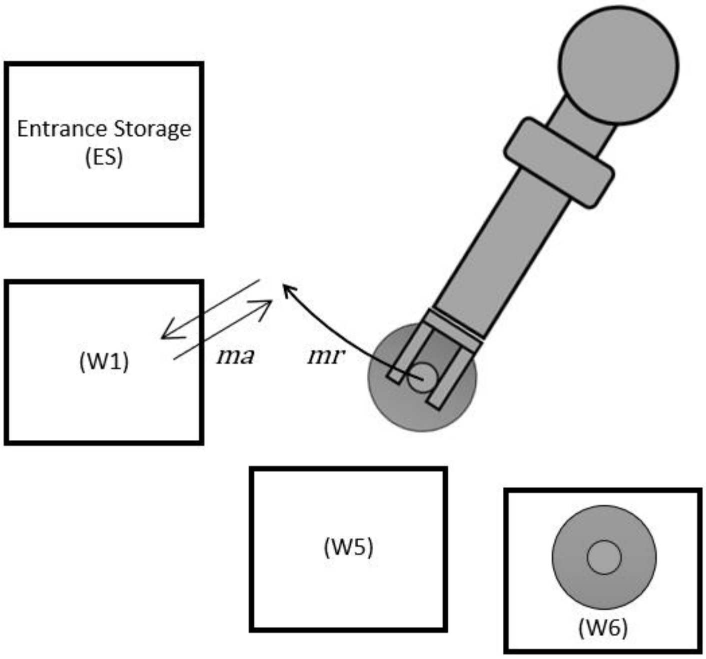

Figure 3. 15: Movement time between W5 and W1

$$M_{ii'} = ma_{i'} + mr_{ii'} \quad (3.6a)$$

$$Mp_{pi} = mr_{pi} + me_i \quad (3.6b)$$

$$mr_{ii'} = mr_{ph} + mr_{hi} \quad (3.6c)$$

With parallel tasks, the rotation $mr_{ii'}$, between Ws (i) and (i') is direct, but the existing prototype returns to the home position h in every movement, so in this case, the rotational motion ($mr_{ii'}$) in the equation 3.6a is defined by the equation 3.6c.

- The rotation motion between several workstations was observed, the average speed v of the rotation time, varies slightly depending on the movement, since the speed is low and the acceleration from zero to this speed is almost instant, the change in velocity from 0 to average speed is considered instantaneous. With this in consideration, the rotation time between Ws can be calculated with equations 3.7, that make sure that the robot does not cross its dead spot:

$$m_r = \begin{cases} \dfrac{\left| |\angle p - \angle i| - 2\pi \right|}{v} & (|\angle p - \angle i| > \pi \ \wedge \ \neg(\min(\angle p, \angle i) < \angle dp < \max(\angle p, \angle i)) \vee \\ & (|\angle p - \angle i| \leq \pi \wedge \min(\angle p, \angle i) < \angle dp < \max(\angle p, \angle i)) \quad (3.7a) \\ \dfrac{|\angle p - \angle i|}{v} & otherwise \quad (3.7b) \end{cases}$$

With $\angle p$, $\angle i$, $\angle dp$ being the current workstation angle, next workstation angle and dead spot angle respectively relative to the home position axis, from table 3.2.

Equation 3.7a is used if the difference in angles is larger than π, meaning the robot takes the longer path, and the rotation between $\angle p$ and $\angle i$ doesn't cross the dead spot, or if the rotation is smaller than π and crosses the dead spot, the purpose of the equation is to "force" the rotation to be through the opposite side and avoid the dead spot.

Equation 3.7b is for the opposite case, meaning the $\angle p$ to $\angle i$ rotation.

- The approach motion (ma) and the exit motion (me) were observed in the real asset, and take the following time in the table 3.3:

	ES	W1	W2	W3	W4	W5	W6	W7	W8	ES
ma (s)	5.0	14.5	10.0	4.0	4.0	3.0	10.0	3.0	3.0	18.0
me (s)	5.0	14.0	17.0	4.0	4.0	13.0	16.0	3.0	3.0	4.0

Table 3. 3: Approach and exit motion times of the workstations

With the data collected and a chosen rotation speed, defined in section 4.1, it is possible to define both all the approach motion times (ma), the rotational motion times (mr) and the exit motion times (me), in the tables 4.2, 4.3 and 4.4 respectively, with all the possible movements, with a defined average speed.

3.3.2 Workstations Time Study

Most processes are either stochastic, or can change in real-time, but there is the exception in some of them, in W2, the quantity of each and different products depends on the customer's request, for these reasons a deeper analysis was done onto the workstation, with the following results:

- There are 3 nozzles in a line configuration, each with a different solution, if there are more than one solution, the nozzle must slide relative to the bottle, to align the correct nozzle, the mean time it takes to move between nozzles is shown in table 3.4:

Nº of movements between nozzles	**0**	**1**	**2**
Time (s)	9	13	16

Table 3. 4: Time to change nozzle in W2

Because its unknown the location of the liquids, the time to move between one or two workstations is considered as the average of the two, that is 14.5 seconds, this is compensated by adding stochasticity to the process.

- The time it takes to fill a recipient naturally depends on the amount of liquid solution, and since the quantity of liquid is measured with a scale, the liquid flow diminishes when the weight is close, taking longer to fill to the desired amount which adds variability to the process, for calculations, the flow rate of the three volumetric pumps is assumed to be 50% of the nominal flow rate. The recipient was filled with different quantities to obtain a curve that represents that relation, shown in figure 3.16,

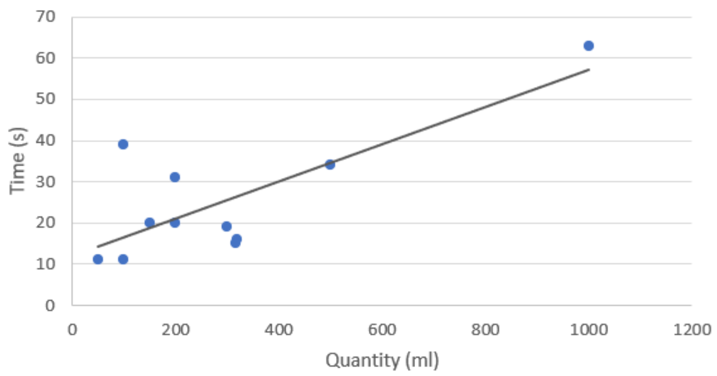

Figure 3. 16: Plot of time to fill a bottle with solution, depending on the quantity, real data is represented by the blue dots, and the gray line is the approximation curve

The chosen curve's equation is defined by tf_l as the time to fill with liquid l with a q_l quantity that represents the relation is:

$$tf_l = 0.045 * q_l + 11.49 \quad (3.8)$$

Since there is a large variation of data relative to the average curve, a new curve is necessary that defines the distribution of occurrences relative to the average, so the data was converted into percentiles to then find a curve fitting that can describe the variation (figure 3.6).

The difference from the expected value (Time/Quantity Curve) and the acquired data was determined, and both the average (μ) and the standard deviation (σ) were calculated, with values of 0.63 and 9.36 respectively, then the probability density function $f(x)$, utilized for normal distribution, was applied to the resulting data (figure 3.5).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.9)$$

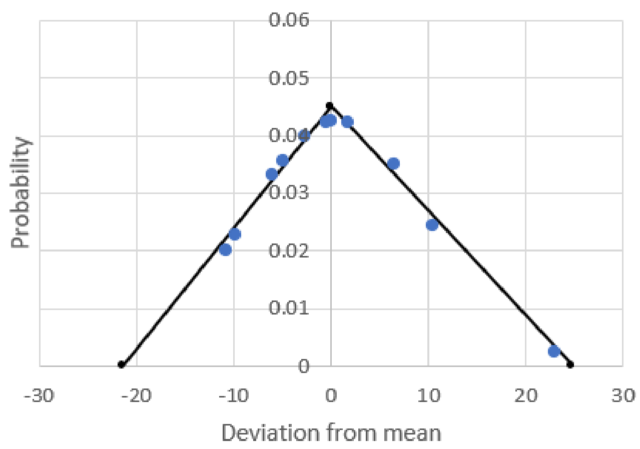

Table 3. 5: Probability density function of the time in W2, blue dots are the percentiles of the real data, and the black line is the chosen fitting distribution

With the percentiles calculated, a fitting that describes the data points can be determined, from a graphical standpoint (figure 3.6), the triangular distribution (eq. 3.10) seems to describe the data with enough accuracy, and so, this fitting was chosen.

$$f(x) = \begin{cases} 0 & for\ x \leq -21.36 \\ 0.0021x + 0.045 & for\ -21.36 < x < 0 \\ 0.045 & for\ x = 0 \\ -0.0018x + 0.045 & for\ 0 < x < 24.75 \\ 0 & for\ x \geq 24.75 \end{cases} \quad (3.10)$$

The time it takes to fill a bottle with liquid can be defined by the following equation with n as the total number of liquids and ts_l is the time to move between nozzles from the comments of table 3.4, with a sample s from the probability distribution $f(x)$ is added to represent stochasticity but does not come from recipe. The time is defined by the equation 3.11:

$$tw2 = \sum_{l=0}^{n} (tf_l + ts_l) + s(f(x)) \quad (3.11)$$

To make use of the equations to obtain the time in the W2, a list of customer requests is necessary, with this purpose in mind, entities involved in the development of the asset were consulted, the main recommendations were:

- There is usually one solution in larger volume than others, for example, solution one is 500 ml and solution two is 100 ml.
- To create a large list of possible requests and take a sample for simulation.

With this feedback, a sheet of 500 random customer requests was created, with 300 requests containing three products and the rest containing two.

In the case of three products, the quantity of the main one q_l in ml is a random value following the discrete uniform distribution over the set {400, 500, … ,900, 1000}, the second product is also a value following the same type of distribution over the set {100, 200, 300, 400}, for the third one, the set of values is {50, 60, …, 190, 200}. For a mixture of two products, the value of the largest quantity follows the same distribution and set, the second product, follows the same distribution from a set {50, 60, …, 390, 400}. The final list of 500 customer requests can be consulted in Appendix B.

For W3, experts claimed that the processing time depends on the quantity of liquid in W2, it takes an average of 5 minutes to mix, and for small quantities, this time is reduced by 1-2 minutes, the time is increased by the same amount for larger quantities, with this data, a simplified curve was created (figure 3.6):

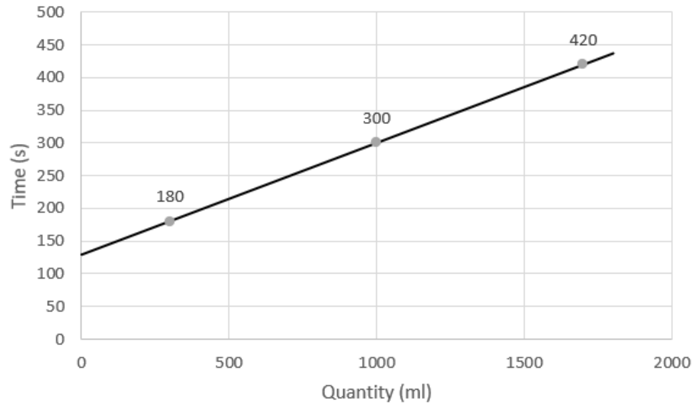

Table 3. 6: Plot of time to mix a solution in W3, depending on the quantity of liquid

With an equation t_{mL}, with L as the total amount of liquid:

$$t_{mL} = 0.171 * \sum_{l=1}^{l=3} (q_l) + 128.57 \tag{3.12}$$

And so, the processing time from recipe for W3 was then added sheet of customer requests, depending on the quantities in W2.

For W3 and W6, in which the process time can be changed in real-time, experts in the field advised to multiply the processing time from recipe by a triangular probability distribution $fr(x)$ defined by equation 3.13:

$$fr(x) = \begin{cases} 0 & for\ x \leq 0.5 \\ 0.5x + 1 & for\ 0.5 < x < 1 \\ 0.045 & for\ x = 1 \\ -0.5x + 1 & for\ 1 < x < 1.5 \\ 0 & for\ x \geq 1.5 \end{cases} \tag{3.13}$$

The W4 (pH check) has two stages, in the first one (W4 1), the pH is checked following the triangular probability distribution $fr(x)$, and if the pH value is incorrect, it goes to the second stage (W4 2) where the pH is corrected.

As for the rest of the workstations, the workflow is linear and does not depend on multiple parameters, the time data was obtained either by observation or by consulting entities involved in the development of the prototype, the data is divided in two tables, the first table 3.7, shows the times of the current system by observation, useful for validation. The second table 3.8, contain data that experts want to test and expected to be requested by clients, combined with data from observation.

Workstation	W1	W2	W3	W5	W1 2	W6	W7	W1 3	W8
Time (s)	27	117	22	30	13	9	51	15	27

Table 3. 7: Processing times in workstations with data collected from observation

Workstation	W1	W2	W3	W4 1	W4 2	W5	W1 2	W6	W7	W1 3	W8
Time (s)	29	*	*	10	25	30	13	300	51	15	27

Table 3. 8: Processing times in workstations, with data collected from observations and from experts in the field, with the (*) symbol representing variable data.

Table 3.9 was then combined with the data from the W2 and W3 to create a list with 500 customer requests, of which, a sample can be seen in Appendix B.

3.3.3 Process Decisions

Decisions such as the need for a bottle to check pH and labelling come from recipe in preparation requests, although real costumer requests are not available, individuals involved in the development can be consulted, regarding the need to do these two processes, the following information was obtained:

- In respect to the pH check, about 50% of requests need to go through this process, and out of the ones that do, 40% need to do it again.

  To address this, 500 random values from a discrete uniform distribution with a set:{0,1} were picked, with one meaning that pH correction is needed and vice versa, with the goal of creating a list of possible requests.

  Also in the model, a poisson distribution with a rate of occurrence of 0.9 was utilized, in which around 40% of the area of the function is larger than one, allows to choose how many times a bottle needs to go to W4 and emulate worker decisions.

- Regarding labelling, since the labelling machine can run out of labels, the system must still be able finish the jobs, and so 70% of recipes are chosen to need a label, even though every bottle should have one. So, 500 random values from a skewed distribution with a set:{0,1}, that allows 70% of the of the requests to have labelling, were added. This list can be seen in Appendix B.

Chapter 4

Simulation Study

The processes of model building, model translation, verification, validation and calibration are not linear, it is necessary for the modeler to come back repeatedly to previous processes, as both corrections and new ways of understanding the real system are an ongoing process.

With the Digital Twin model developed, the next step is to do forecasts for the manufacturing system to understand where it can be improved, using the DT as a simulation model.

4.1 Verification, Validation, and Calibration

The verification was done by analyzing the simulation runs, step by step, through visual validation of the crane movement, checking if the parameters and variables are correct, and manually doing the math related with scheduling to check if the right decisions are being made by the model.

To calibrate and validate the model as a whole, the system can be seen as input-output transformations, where the model receives input parameters and transforms it in output measures of performance, testing the model's capability of predicting the future behavior (output) of the real system, when fed the same input data as in the real counterpart. With this purpose in mind, historical data can be used for these purposes, by comparing the model predictions (outputs) with the results of the real system.

Usually, the model is measured under a range of different input conditions, if the system is still in planning stages, no data can be collected and so, input output validation cannot be used to its full extent, and so calibration and validation of parts of the system with available data can be conducted. In this case study, only the part of the system shown in figure 4.1, has useful data for this purpose.

In order for the workflow to be applicable, there needs to be proof that the data and workflow processed in the Digital Twin are faithful to the real asset. A good way to address this necessity, is to emulate the workflow of the existing system and check if the processes occur in the right order and that they finish at a similar time, figure 4.1 shows a diagram of the existing system.

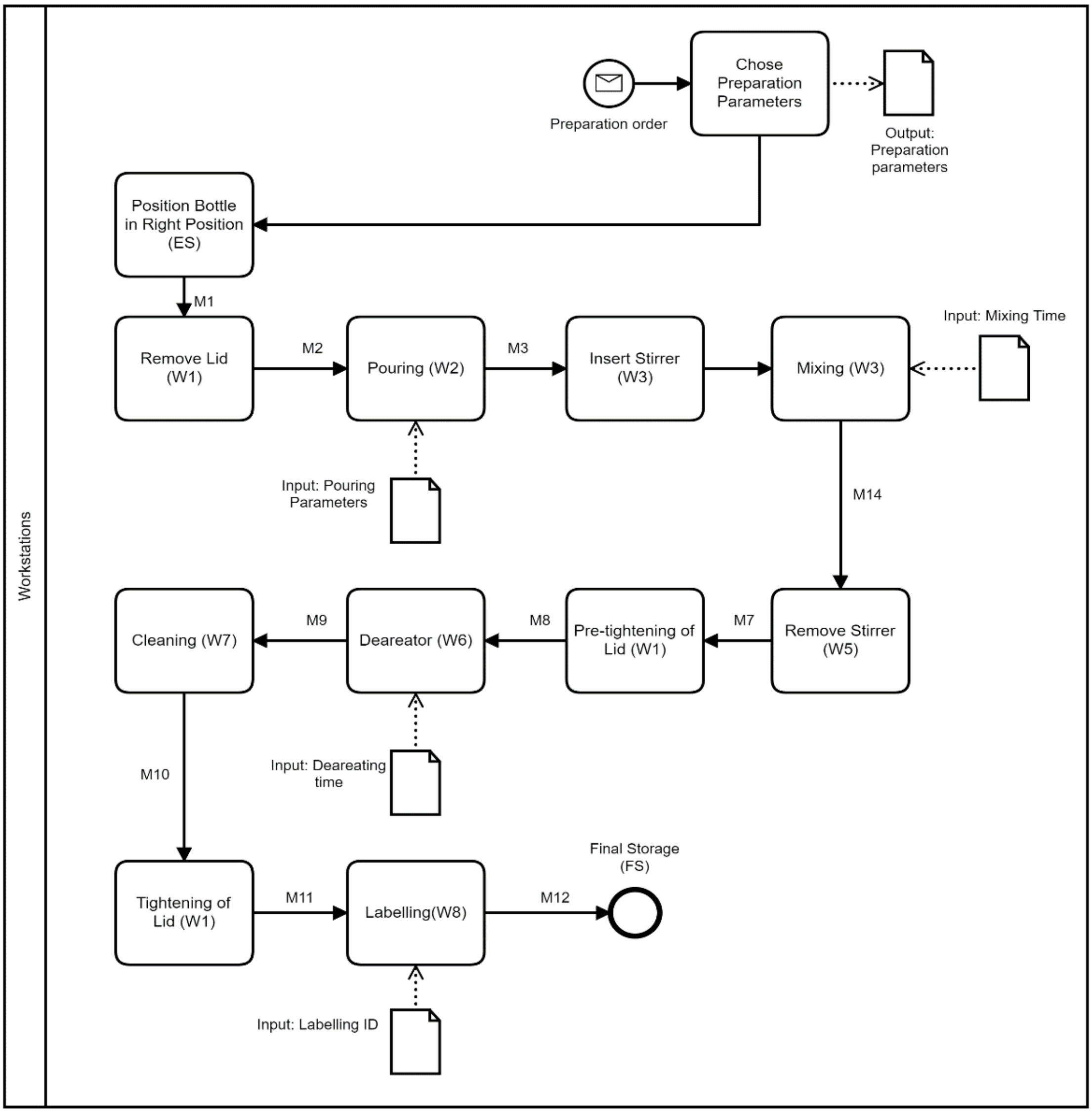

Figure 4. 1: Process workflow for model calibration and validation

To see if the data is faithful to the real system and the model works correctly, the “Parameter Simulation Tool” in Anylogic was utilized, this tool gives the capability to test different parameters with a chosen number of replications, in the case of validation, parameters are fixed, only one replication is necessary, as it is only intended to compare the time to the real counterpart and so, stochasticity is not considered, only the data from table 3.8, containing the processing times.

Since the complex robot movement was simplified in three parts, explained in section 3.3.1, and the rotation speed is not constant, an acceptable rotation speed (v), for the rotation time (mr), must be chosen for calibration of the model.

This speed must result of the compromise off all the movements, in a way that the movement times of the Digital Twin closely emulate the actual times. With this purpose in mind, multiple rotation speeds from 3 rad/s to 4 rad/s were tested, and the relative approximation error E_{iv} at the end of the operations (i) and accumulated relative approximation error E_v of all operations (r) of the model time relative to the actual time, defined by equation 4.1a and 4.1b respectively were used as measure of performance, with the results in figure 4.2.

$$E_{iv} = \frac{|Ts_{iv} - Tr_i|}{Tr_i} * 100 \tag{4.1a}$$

$$E_v = \sum_{i=1}^{r} \left(\frac{|Ts_{iv} - Tr_i|}{Tr_i} * 100 \right) \tag{4.1b}$$

Tr_i: Real-time at which process i finishes.

Ts_i: Simulation time at which process i finishes for rotation velocity of v.

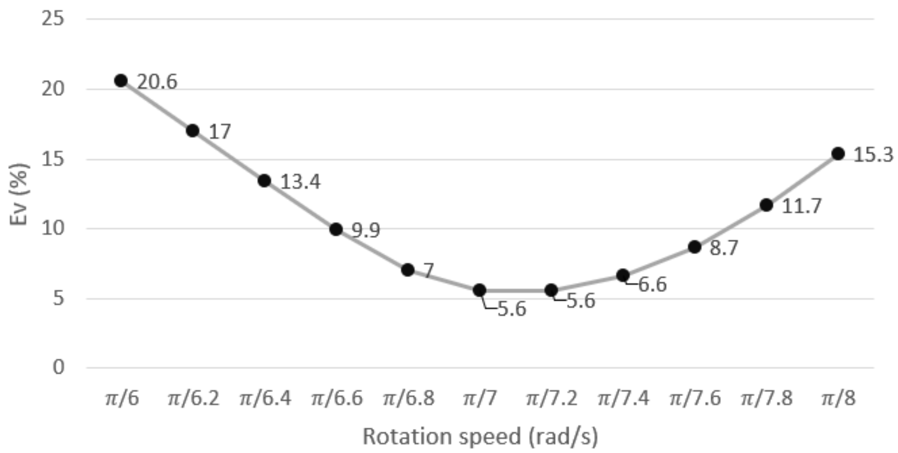

Figure 4. 2: Plot of Accumulated relative approximation error, depending on the rotation speed

With testing results from different rotation speed values, it was shown that $\pi/7.2$ rad/s was the value that ensures the smaller accumulated error of 5.6%, and so, it was the chosen speed for calibration and validation. The comparison between completion time from validation and the Digital Twin can be consulted in figure 4.3, and table 4.1 shows the relative approximation error of each operation:

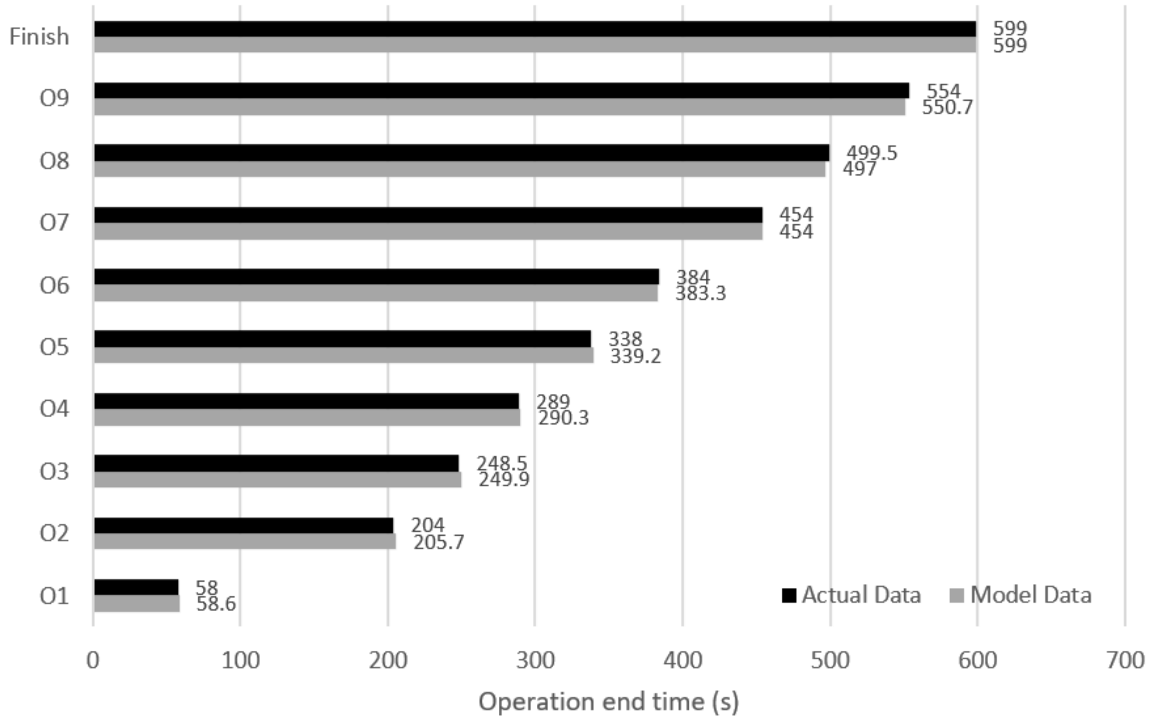

Figure 4. 3: Plot of completion time of processes from validation (black), compared to the Digital Twin (gray), with "O1" representing the completion time on the first operation in W1 until the bottle reaches the FS on "Finish"

	O1	O2	O3	O4	O5	O6	O7	O8	O9	Finish
E_{iv} (%)	1.03	0.83	0.56	0.46	0.34	-0.17	-0.01	-0.50	-0.59	-1.04

Table 4. 1: Relative approximation error, with "O1" representing the completion time on the first operation in W1 until the bottle reaches the FS on "Finish"

With a maximum value of relative approximation error of 1.04%, although this approximation is not as valuable as having real data regarding all the movements, it can describe the movement times for the existing system with enough accuracy. The definitive tables 4.2, 4.3 and 4.4 containing the Rotation (t_{rpi}), pre movement (t_{pij}) and movement times ($t_{ii'}$), with all the possible movements, where then defined.

$mr_{ii'}$(s)	ES	W1	W2	W3	W4	W5	W6	W7	W8	ES	Home
ES	0.00	1.36	4.24	3.04	5.44	2.00	10.80	9.20	8.36	7.20	4.24
W1	1.36	0.00	5.60	4.40	6.80	0.64	12.16	10.56	9.72	8.56	5.60
W2	4.24	5.60	0.00	1.20	1.20	6.24	6.56	4.96	4.12	2.96	0.00
W3	3.04	4.40	1.20	0.00	2.40	5.04	7.76	6.16	5.32	4.16	1.20
W4	5.44	6.80	1.20	2.40	0.00	7.44	5.36	3.76	2.92	1.76	1.20
W5	2.00	0.64	6.24	5.04	7.44	0.00	12.80	11.20	10.36	9.20	6.24
W6	10.80	12.16	6.56	7.76	5.36	12.80	0.00	1.60	2.44	3.60	6.56
W7	9.20	10.56	4.96	6.16	3.76	11.20	1.60	0.00	0.84	2.00	4.96
W8	8.36	9.72	4.12	5.32	2.92	10.36	2.44	0.84	0.00	1.16	4.12
ES	7.20	8.56	2.96	4.16	1.76	9.20	3.60	2.00	1.16	0.00	2.96
Home	4.24	5.60	0.00	1.20	1.20	6.24	6.56	4.96	4.12	2.96	0.00

Table 4. 2: Rotation times of all possible movements between workstations including the home position

Mp_{pi} (s)	ES	W1	W2	W3	W4	W5	W6	W7	W8	ES	Home
ES	5.00	15.36	21.24	5.04	9.44	15.00	26.80	12.20	11.36	11.20	4.24
W1	6.36	14.00	22.60	6.40	10.80	13.64	28.16	13.56	12.72	12.56	5.60
W2	9.24	19.60	17.00	3.20	5.20	19.24	22.56	7.96	7.12	6.96	0.00
W3	8.04	18.40	18.20	2.00	6.40	18.04	23.76	9.16	8.32	8.16	1.20
W4	10.44	20.80	18.20	4.40	4.00	20.44	21.36	6.76	5.92	5.76	1.20
W5	7.00	14.64	23.24	7.04	11.44	13.00	28.80	14.20	13.36	13.20	6.24
W6	15.80	26.16	23.56	9.76	9.36	25.80	16.00	4.60	5.44	7.60	6.56
W7	14.20	24.56	21.96	8.16	7.76	24.20	17.60	3.00	3.84	6.00	4.96
W8	13.36	23.72	21.12	7.32	6.92	23.36	18.44	3.84	3.00	5.16	4.12
ES	12.20	22.56	19.96	6.16	5.76	22.20	19.60	5.00	4.16	4.00	2.96
Home	9.24	19.60	17.00	3.20	5.20	19.24	22.56	7.96	7.12	6.96	0.00

Table 4. 3: Pre-movement times of all possible movements between workstations including the home position, the robot starts the movement on the workstations in the bottom, and finishes in the workstations in the right

$M_{ii'}$(s)	ES	W1	W2	W3	W4	W5	W6	W7	W8	ES	Home
ES	5.00	15.86	14.24	7.04	9.44	5.00	20.80	12.20	11.36	39.20	4.24
W1	6.36	14.50	15.60	8.40	10.80	3.64	22.16	13.56	12.72	40.56	5.60
W2	9.24	20.10	10.00	5.20	5.20	9.24	16.56	7.96	7.12	34.96	0.00
W3	8.04	18.90	11.20	4.00	6.40	8.04	17.76	9.16	8.32	36.16	1.20
W4	10.44	21.30	11.20	6.40	4.00	10.44	15.36	6.76	5.92	33.76	1.20
W5	7.00	15.14	16.24	9.04	11.44	3.00	22.80	14.20	13.36	41.20	6.24
W6	15.80	26.66	16.56	11.76	9.36	15.80	10.00	4.60	5.44	35.60	6.56
W7	14.20	25.06	14.96	10.16	7.76	14.20	11.60	3.00	3.84	34.00	4.96
W8	13.36	24.22	14.12	9.32	6.92	13.36	12.44	3.84	3.00	33.16	4.12
ES	12.20	23.06	12.96	8.16	5.76	12.20	13.60	5.00	4.16	32.00	2.96
Home	9.24	20.10	10.00	5.20	5.20	9.24	16.56	7.96	7.12	34.96	0.00

Table 4. 4: Movement times of all possible movements between workstations including the home position, the robot starts the movement on the workstations in the right, and finishes in the workstations in bottom

4.2 Simulation Runs

With the model validated, the next step is to do simulation runs with multiple bottles and a variety of different parameters, which are called iterations, and discover what decisions reduce the makespan and increase the utilization of the equipment.

In the model with parallel tasks, characteristics such as the stochasticity in processes and real-time decisions, make single simulation runs not statistically representative, and demands a large number of replications, which are the repetition of simulation runs where only stochastic parameters are changed, utilized in distribution samples [52], in this context, the stochastic parameters are:

- Number of times in the W3-W4 cycle and decision if it is necessary to do the correction, that is the W4_2 process.
- Stochasticity that affects process times, defined in Chapter 3, Data Processing.
- Order choice, picking a random order following the discrete uniform distribution over the set {1, 2, … ,499, 500}.

These variables make the performance measurement a difficult process as this variation also affects decision making of the algorithm as entities finish processes at different times. The inherent stochasticity makes the comparison of different single iterations unreliable.

As running simulations takes time, choosing a number of replications that is too high, would limit the amount of possible combinations to test, as the modeler has limited time to do simulation runs.

To address this, another Parameter Variation simulation study was performed, in order to determine the adequate amount of replications to have a significative sample size, because a different amount of total orders or number of bottles will be tested for the effect on the completion time, the chosen measurement criteria was Relative Margin of Error to the sample mean (R_{MOE}) as a selected R_{MOE} is valid independently of the number of bottles.

With a confidence level (CI) of 95% (eq. 4.2a). There is 95% probability that sample p is within the 95% CI [53].

$$R_{MOE_{95\%}} = \frac{Bu - Bl}{\mu} * 100 \quad (4.2a)$$

Bu – Upper bound of a 95% confidence interval

Bl – Lower bound of a 95% confidence interval

μ – Sample mean

The bounds of the 95% confidence intervals, can be calculated with the formula:

$$B = \mu \pm z \frac{s}{\sqrt{n}} \quad (4.2b)$$

B – Bound of a 95% confidence interval, if the $\pm$ in the equation is positive, it’s the upper bound, otherwise, it’s the lower bound.

s – Sample standard deviation

z – Chosen z value, 1.96 for a 95% confidence interval

A number to 50 bottles were chosen for testing, an error <2% was selected for accuracy, to allow for an accurate measure of performance and to compare different model runs.

With parallel tasking, and the SPT dispatching rule, for 50 bottles, the following figure 4.4 shows the $R_{MOE_{95\%}}$ for different number of replications:

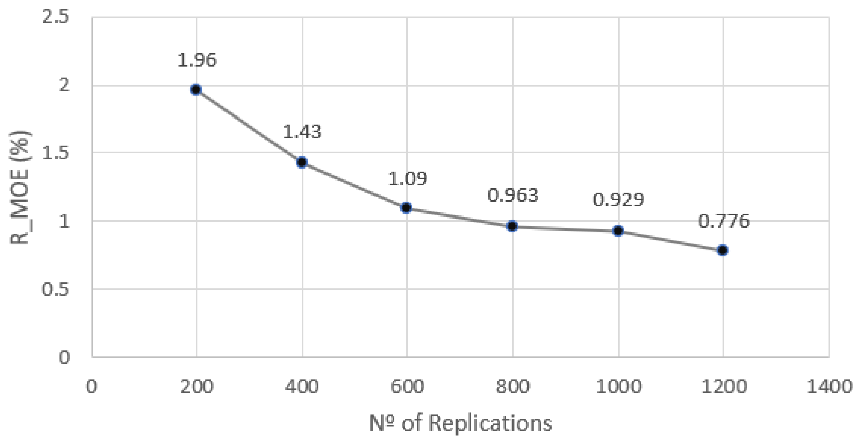

Figure 4. 4: Plot of relative margin of error to the sample mean, dependent on the number of replications

To have an acceptable error <1%, 1000 replications were decided as an adequate value, as the variation for 800 replications might be larger than one depending on the different experiments applied on the system. If the $R_{MOE_{95\%}}$ is larger than 1% with different experimentations, a higher number of replications are to be employed.

4.3 Results and Analysis

In this step, different model parameters are chosen to better understand where the system can be improved, for those, different measures of performance are utilized:

Makespan (T): time to process all the Jobs, or total time.

Resource Utilization (U_w): measures the utilization of a Workstation or the Robot, is the relation between the total working time of a resource (Tu_W) and the makespan (eq. 4.3):

$$U_w = \frac{Tu_w}{T} \quad (4.3)$$

Resource Occupation (Ro): measures the occupation of a workstation, that is, the percentage of time, these have bottles, independent of the working time, is the relation between the occupation time of a resource (To_w) and the makespan (eq. 4.4):

$$O_w = \frac{To_w}{T} \quad (4.4)$$

Performance Improvement (I): also called makespan reduction or reduction in total completion time, is to compare the makespan of different iterations, and measure the effect of parameter changes, with T_p as the makespan of the iteration used as a comparison term, (eq. 4.5).

$$I = \left(1 - \frac{T}{T_p}\right) * 100 \quad (4.5)$$

For each change in configurations and parameters, only the dispatching rules that has the lowest makespan are shown, in Appendix A, more detailed results and analysis can be consulted to visualize the measurements of performance of other dispatching rules and configurations, as well as tables containing utilization/occupation information.

4.3.1 Parallel Tasks

The prototype currently only supports single tasking, where a new job is only processed when the last one is completed, as said in the introduction, one of the goals of this thesis is to see the effect of parallel tasking in the completion time, with different heuristics. For this, an iteration with single tasks was compared with the parallel tasking with the dispatching rules referred in Chapter 3s Scheduling and Algorithm, with 50 bottles (figure 4.5):

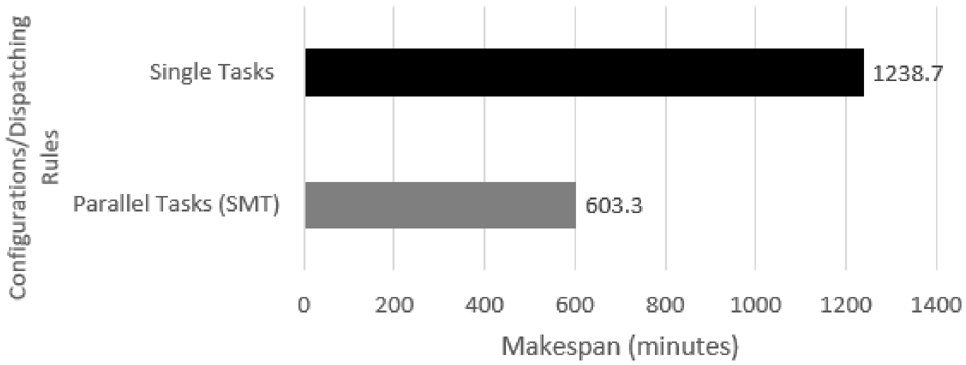

Figure 4. 5: Makespan comparison of single tasking with parallel tasking with the best performing SMT rule

Combining parallel tasking with the SMT dispatching rule originates the best results, reducing the overall completion time by 51.3%, when comparing with single tasking, making this change imperative to improve the system, as well as allowing the system to be improved by adding identical parallel machines in key processes.

4.3.2 Home Position

Individuals involved in the development of the prototype claimed the robot goes to the home position for safety reasons, as it's a prototype, but they claim that in the future its intended for the need to go to the home position in not necessary, and so, the model was put to test to determine the effect of the home position in the performance (figure 4.6):

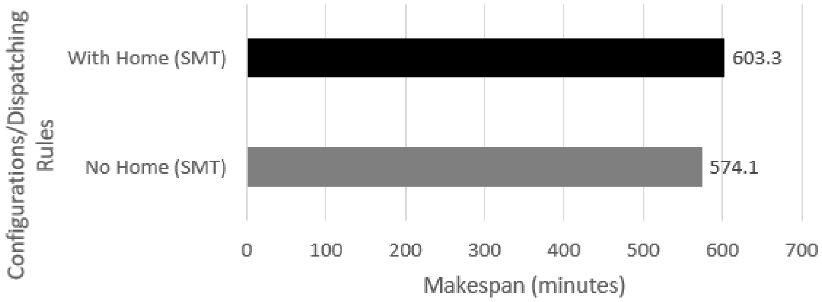

Figure 4. 6: Makespan comparison of having to go to home position (black), with the SMT rule, and not going to the home position (gray), with SMT rule

To better understand where the model can be improved, various experiments with different parallel machine configurations were ran for the different dispatching rules, the table 4.5 shows information relative to workstation and robot utilization and occupation for the model with single machines for the SMT dispatching rule as it is the one with the lowest makespan.

Resource	Utilization (%)	Occupation (%)
Robot	51.1	-
W1	8.3	40.8
W2	16.7	82
W3	78.8	85.8
W4	3.1	20.2
W5	4.1	16.1
W6	43.7	50.7
W7	7.4	17.5
W8	1.9	4.2

Table 4. 5: Utilization/occupation table of the single machines configuration with the SMT rule (No home)

After testing analyzing table 4.5 and figure 4.6, it was shown that apart from the CSMT, the dispatching rules didn't have a large effect on the utilization of resources and makespan. Going to the home position does not affect the process time dramatically, with a reduction in makespan of 4.8%.

This is because, most of the time, no more than one bottle is available, which is proven by the low utilization of the robot, as its idle almost half the time, due to the necessary availability conditions restricting the possible movements, and the large time gap between short and long processing times, which creates bottlenecks, proved by the utilization gap between Ws, of long processes (W3, W6) and short ones (W1,W4, W7, W8).

4.3.3 Resource Allocation

The W3 has the largest utilization and occupation rate, and its predecessors in the workflow, as seen in figure 3.7: W2 and W4, have a large difference between its occupation and utilization rates, implying that the bottles at these Ws spend most of the time waiting, which indicates a possible bottleneck at W3. The makespan of the configuration with an extra W3 with different dispatching rules can be seen in the figure 4.7:

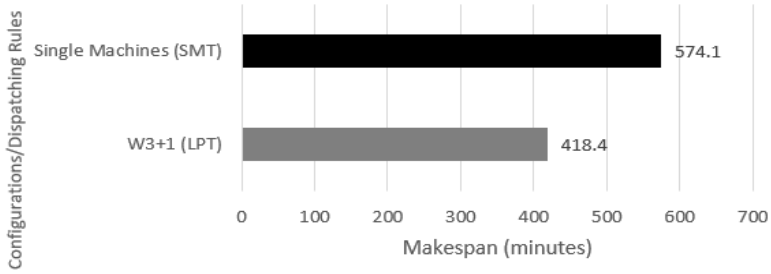

Figure 4. 7: Makespan comparison of having single machines, with the SMT rule, and having an extra W3 (gray), with the best performing LPT rule

As seen on figure 4.8, adding a parallel W3 yields an increase in performance of about 27.1% when comparing with the best performing dispatching rule with single machines, meaning the W3 is a bottleneck, applying this change in the real counterpart proves effective, with the LPT algorithm being the most effective for the configuration, the following table shows the utilization/occupation of the new configuration (table 4.6):

Resource	Utilization (%)	Occupation (%)
Robot	72.6	-
W1	11.4	57.3
W2	22.8	44
W3	54.2	68.9
W4	4.3	19.4
W5	5.6	48.1
W6	59.6	69.8
W7	10.2	91
W8	2.6	7.8

Table 4. 6: Utilization/occupation table with configuration containing two W3s, with the LPT rule

With this new configuration, the W3 might still be a bottleneck as its utilization/occupation ratio is still high, W6 might also be another possible bottleneck, as it has both a large utilization and occupation, W1, with a large occupation and having each job going through it, makes considering an extra parallel W1 a possible improvement to the system, the result with all the heuristics can be consulted in table A.3.

Combinations where then tested based on the analysis, to determine where the system can be improved, the combinations bellow offer the best results (figure 4.8):

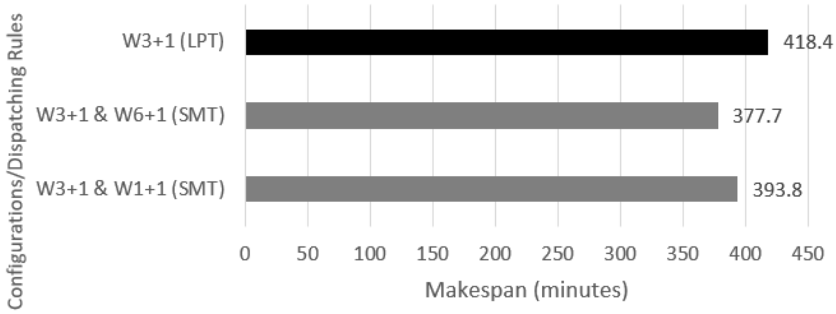

Figure 4. 8: Makespan comparison of having two W3s, with the LPT rule, with having an extra W3 with an W6 or W1 (gray), with the best performing SMT rule

According to the Digital Twin, adding an extra machine to W6 when already having two W3s, further decreases makespan by about 9.7%.

Adding an extra W1, when already having two W3s, results in a performance increase of 5.9%, it might not be as effective as adding a W3 and a W6, but depending on the cost of the machines it might be a viable option, table 4.7 then shows the utilization/occupation of the configuration with an extra W3 and W6.

Resource	Utilization (%)	Occupation (%)
Robot	79.3	-
W1	12.6	50.3
W2	25.2	63.8
W3	60.2	74.5
W4	4.7	24.7
W5	6.2	46.3
W6	33.1	42.9
W7	11.3	32.8
W8	2.9	7.7

Table 4. 7: Utilization/occupation table with configuration containing two W3s and two W6s , with the SMT rule

W3 still possesses the highest utilization, so its still restricting the workflow and W6 has still the 2nd overall highest utilization.

W1 appears to restrict the flow of tasks, as W5 has a low utilization/occupation ratio, since each job need to go through W1, having a single machine might create a choke point.

As the combination of extra parallel W3 and W6 has the lowest makespan for most dispatching rules, it is utilized as a comparison term to other configurations (figure 4.9):

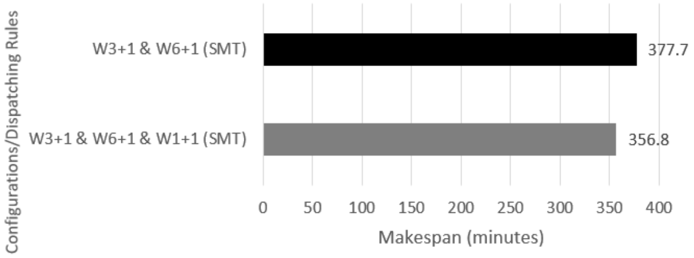

Figure 4. 9: Makespan comparison of having two W3s and W6s, with the SMT rule, with having two W3s, W6s and W1s (gray), with the best performing SMT rule

Including an extra W1 to the configuration that already possesses two W3s and W6s, yields the largest reduction in makespan of 5.5%, on top of this configuration an extra W3 was added, with the following results (figure 4.10):

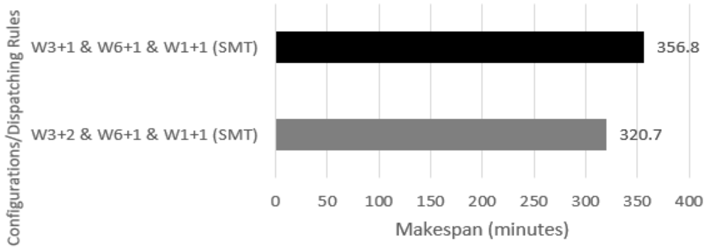

Figure 4. 10: Makespan comparison of having two W3s, W6s and W1s, with the SMT rule, with having three W3s and two W6s and W1s (gray), with the best performing SMT rule

Adding an extra W3, relative to previous configuration, further reduces the makespan by 10.1%, other additions were tested, using this configuration as a reference, but the makespan does not decrease more than 2%, so adding more Ws does not yield much better results, the utilization and occupation can be seen in figure 4.8:

Resource	Utilization (%)	Occupation (%)
Robot	92.6	-
W1	7.4 /71.4	71.4
W2	29.8 /61.4	61.4
W3	47.1 /66	66
W4	5.6 /35.1	35.1
W5	7.3 /62.4	62.4
W6	38.9 /61	61
W7	13.3 /37.7	37.7
W8	3.4 /9.3	9.3

Table 4. 8: Utilization/occupation table with configuration containing three W3s and two W6s and W1s , with the SMT rule

The robot is now utilized most of the time (92.6%), with its high utilization, the movement time might significatively be hindering the workflow, as the workstations still have an overall low utilization, and increasing the number of parallel machines no longer yields reductions in makespan larger than 2%, also the fact that the SMT rule, which relates with the robot’s speed, is performing better than other dispatching rules, makes a study on the robot movement speed a relevant topic.

4.3.4 Robot Speed

The robot currently works at a slow speed as it is only a prototype, but it can be increased further down the line, and so, experimenting with the robot speed for various resource allocations might reduce the makespan considerably.

The configurations that have the largest impact on the makespan as well as the single machines configuration, were employed and a factor of 1.25 and 1.5, was multiplied by both the pre movement and movement time for this experiment, with the following results:

Standard configuration (figure 4.11):

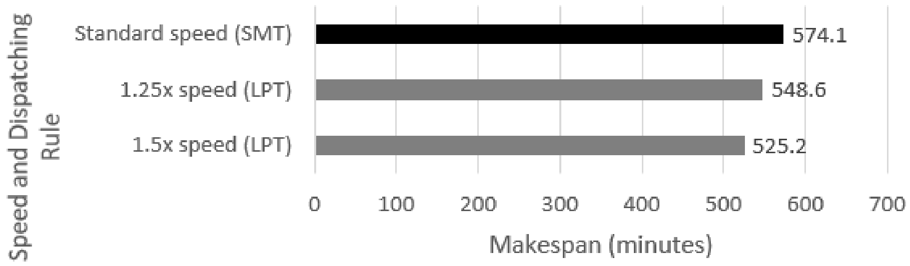

Figure 4. 11: Makespan comparison of increasing the robot speed for the single machines configuration

Two W3s (figure 4.12):

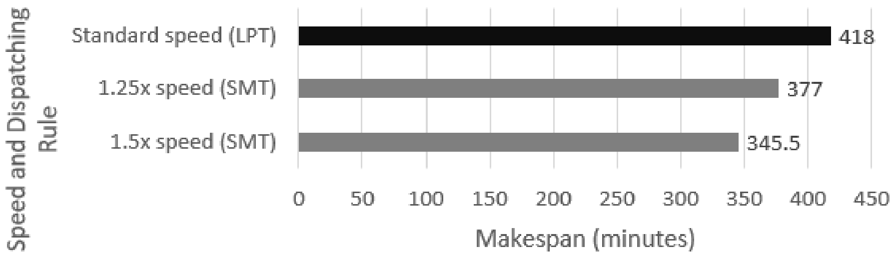

Figure 4. 12: Makespan comparison of increasing the robot speed for the configuration with an extra W3

Two W3s and W6s (figure 4.13):

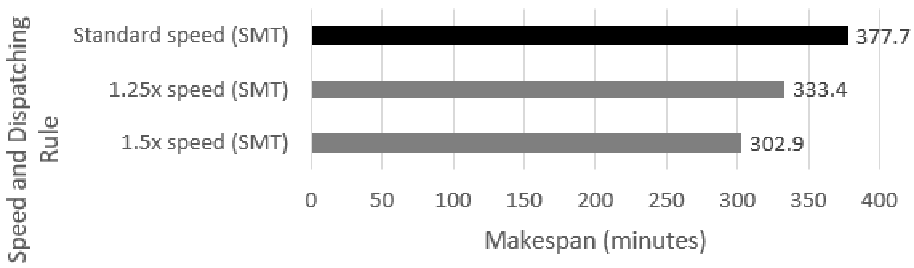

Figure 4. 13: Makespan comparison of increasing the robot speed for the configuration with an extra W3 and W6

Two W3s, W6s and W1s (figure 4.14):

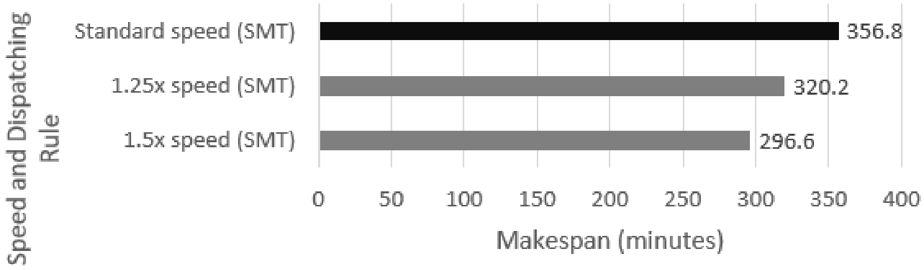

Figure 4. 14: Makespan comparison of increasing the robot speed for the configuration with an extra W3, W6 and W1

Three W3s and two W6s and W1s (figure 4.15):

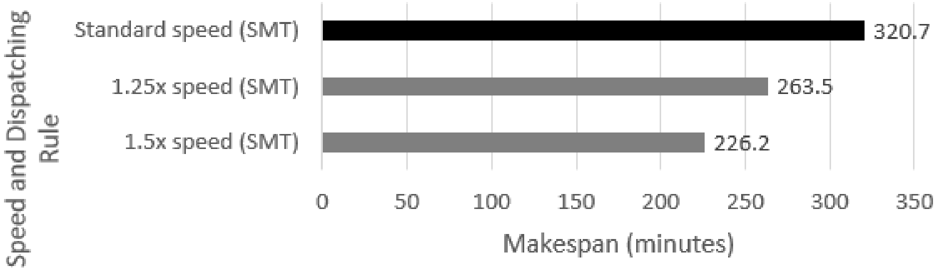

Figure 4. 15: : Makespan comparison of increasing the robot speed for the configuration with two extra W3s, and one extra W6 and W1

To better compare the impact of increasing the robot's speed, table 4.9 was created:

Configuration	Robot 25% faster (%)	Robot 50% faster (%)
Single Machines	4.4	8.5
Two W3s	9.8	17.3
Two W3s and W6s	11.7	19.8
Two W3s, W6s and W1s	10.3	16.9
Three W3s, W6s and W1s	17.8	29.5

Table 4. 9: Makespan reduction for different configurations and robot speeds

Increasing the velocity of the robot does not reduce the single machines configuration's completion time as much as for other configurations, for the same reasons discussed on the 4.3.2 section, but for the other configurations, if the velocity of the robot can be increased by 25% and 50%, the makespan is reduced by at least 9.8% and 20.5% respectively.

As said in the previous section 4.3.3, when adding parallel machines no longer yields considerable improvements, and the robot is utilized most of the time, increasing its velocity is a valuable option, as just a 25% increase, results in a reduction in makespan of 17.8%, and with a velocity increase of 50%, results in a considerable makespan reduction of 29.5%.

Combining an increase in robot speed, with adding key workstations, can be considered as the best option, choosing the right combination depends on the cost and feasibility of the changes, information that the author does not possess.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This dissertation developed a Digital Twin of a flexible manufacturing system, focused on simulation to forecast different scenarios. The simulation recurs to the scheduling of parallel tasks while handling variable customer requests in real-time. This thesis bridges the concept of a Digital Twin working with a human counterpart in a flexible environment, in a human centric vision towards Industry 5.0.

With the feedback from individuals involved in the development of the prototype on every stage of the progression of this paper, workflows of each client request and for scheduling decisions were created using the BPMN, this graphical notation provides an easy and intuitive way to understand the workflows of the system.

A mixed event simulation model was chosen as the best approach, where discrete events, such as the robot being free, trigger scheduling decisions. When no jobs are available, the discrete time clock checks the system until a process finishes and the flow of tasks is resumed.

Real-time changes and different customer requests made the schedule building a difficult task, as it quickly gets obsolete, therefore, completely reactive scheduling was selected. The mixed event simulation model was then developed, capable of handling various customer requests, robot movement speeds, as well as different combinations of machines, it recurs to five different heuristics: SPT, LPT, LWR, SMT, and CSMT, to make scheduling decisions.

A list of clients' recipes, each containing different processing times and workflows based on real requests, was created, from which, random requests were chosen for each simulation run. In respect to real-time changes, such as processing time and workflow changes by workers and the inherit time variation in some processes, these were emulated by using suitable probability distributions.

Simulation runs with different parameters were experimented with 50 jobs, with the results in table 5.1:

Configuration	Standard speed	Velocity increase of 25%	Velocity increase of 50%
Single Tasks	1238.7	-	-
Parallel Tasks (SMT)	603.3	-	-
No Home Position (SMT)	574.1	548.1	525.2
W3+1 (LPT)	418.4	377	345.5
W3+1 & W6+1 (SMT)	377.7	333.4	302.9
W3+1 & W6+1 & W1+1 (SMT)	356.8	320.2	296.6
W3+2 & W6+1 & W1+1 (SMT)	320.7	263.5	226.2

Table 5. 1: Absolute makespan values depending on configuration, best dispatching rule and robot velocity

Applying parallel task scheduling with a SMT dispatching rule, results in a reduction of makespan by 51.3%, when comparing with single task scheduling, making the system more efficient by increasing the utilization of resources.

Experiments with additional parallel identical machines were also conducted, the single addition that proves most effective is adding one W3, which decreases the overall completion time by 27% when compared with the single machines configuration. The configuration with the larger number of machines, contains three identical W3s, two W6s and W1s, in which, the completion time is reduced by 44.1% when comparing with the single machines configuration.

The robot speed was also a topic of analysis, it proves beneficial when parallel machines are added in the configuration with two extra W3 and one extra W6 and W1, increasing its velocity by 50% yields a reduction in makespan of 29.5% compared to standard speed. Combining a 50% increase in velocity with the most complete configuration (W3+2 & W6+1 & W1+1) results in a makespan reduction of 82% when comparing with the base configuration with single tasks and single machines.

Between all heuristics, SMT proved to be the most effective in reducing makespan, with the LPT outperforming the SMT in a few configurations. Therefore, the movement time of the robot is most relevant in the scheduling process and yields the best results.

Overall, the objective of developing a Digital Twin capable of making real-time decisions, was achieved by utilizing completely reactive scheduling. The proposed solution allows doing forecasts with different dispatching rules, various machine configurations and robot speed, with access to a visualization window, for a deeper understanding of the model, giving stakeholders more information about where the system can be improved.

5.2 Future Work

5.2.1 Model Improvement

As presented in the contributions, developing a model that is capable of handling real-time decisions efficiently was one of the objectives, and its still a direction in which there are many opportunities for improvement, these being:

- Creating another new dispatching rule that emphasizes on the shortest transportation time and the longest processing time, as the SMT and LPT were the best performing heuristics. With weights on each of the factors, various simulations runs could be performed in order to determine the ratio that reduces the makespan the most.
- Improve the CSMT dispatching rule , one of the issues of this dispatching rule, is that when the robot waits for long processes, it keeps waiting until the process finishes, independently of any jobs becoming available. An improved CSMT dispatching rule, could take advantage of the DTS clock to do checks on the system, triggering the heuristic's decision process, and interrupting the wait, if another job has higher priority.
- Prioritizing some key workstations or operations might reduce the makespan in some configurations, therefore, using an associated weight with each workstation or operation, for the dispatching rules, might prove beneficial.
- A part of the data collection was done by consulting individuals involved in the development of the prototype, so part of the information regarding processing times, real-time decisions and client requests was approximated, with access to real data in respect to these parameters, the simulation results would be more reliable.

5.2.2 Concept Application

The concept of the Digital Twin focused on simulation was achieved, and the next step is to apply this concept to the real asset, where the input of the Digital Twin wouldn't be historical data, but instead, real-time data from the asset, recurring to sensors and the already existing software. For this, an interface that allows processing times and decisions to be altered in real-time, when an operator decides, instead of probability distributions, that represent the decision, would need to be integrated into the Digital Twin.

The operator should have the option to make scheduling decisions, as told in the introduction, people have problem-solving skills and knowledge of the system, therefore, they should have the ability to override scheduling decisions of the Digital Twin, when they see fit, and not just in key processes, which is a topic that is included in the goals of industry 5.0.

Bibliography

[1] P. Karagiannis, N. C. Zacharaki, G. Michalos, and S. Makris, “Increasing flexibility in consumer goods industry with the help of robotized systems,” *Procedia CIRP*, vol. 86, pp. 192–197, 2020, doi: 10.1016/J.PROCIR.2020.01.039.

[2] M. M. Tseng, Y. Wang, and R. J. Jiao, “Mass Customization,” *CIRP Encyclopedia of Production Engineering*, pp. 1–8, 2017, doi: 10.1007/978-3-642-35950-7_16701-3.

[3] H. A. ElMaraghy, “Flexible and reconfigurable manufacturing systems paradigms,” *International Journal of Flexible Manufacturing Systems 2006 17:4*, vol. 17, no. 4, pp. 261–276, Oct. 2006, doi: 10.1007/S10696-006-9028-7.

[4] D. Telgen, “Grid Manufacturing: A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines,” 2017. doi: 10.13140/RG.2.2.24332.10885.

[5] N. Kousi, C. Gkournelos, S. Aivaliotis, C. Giannoulis, G. Michalos, and S. Makris, “Digital twin for adaptation of robots’ behavior in flexible robotic assembly lines,” *Procedia Manufacturing*, vol. 28, pp. 121–126, Jan. 2019, doi: 10.1016/J.PROMFG.2018.12.020.

[6] T. Coito *et al.*, “A Middleware Platform for Intelligent Automation: An Industrial Prototype Implementation,” *Computers in Industry*, vol. 123, p. 103329, Dec. 2020, doi: 10.1016/J.COMPIND.2020.103329.

[7] S. Vaidya, P. Ambad, and S. Bhosle, “Industry 4.0 – A Glimpse,” *Procedia Manufacturing*, vol. 20, pp. 233–238, Jan. 2018, doi: 10.1016/J.PROMFG.2018.02.034.

[8] “Why Smart Manufacturing? - IEEE Spectrum.” https://spectrum.ieee.org/consumer-electronics/standards/why-smart-manufacturing (accessed Jul. 29, 2021).

[9] Gartner, “Gartner Top 10 Strategic Technology Trends For 2019.” https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2019 (accessed Oct. 07, 2021).

[10] T. Coito, B. Firme, M. S. E. Martins, S. M. Vieira, J. Figueiredo, and J. M. C. Sousa, “Intelligent Sensors for Real-Time Decision-Making,” *Automation*, vol. 2, no. 2, pp. 62–82, 2021, doi: 10.3390/automation2020004.

[11] P. Esben H. Østergaard, “Welcome to Industry 5.0,” *https://www.isa.org/intech-home/2018/march-april/features/welcome-to-industry-5-0*, (accessed Aug. 23, 2021).

[12] Maija. Breque, Lars. de Nul, Athanasios. Petridis, and European Commission. Directorate-General for Research and Innovation., *Industry 5.0 : towards a sustainable, human-centric and resilient European industry.* doi: 10.2777/308407.

[13] C. B. Frey and M. A. Osborne, “The future of employment: How susceptible are jobs to computerisation?,” *Technological Forecasting and Social Change*, vol. 114, pp. 254–280, 2017, doi: https://doi.org/10.1016/j.techfore.2016.08.019.

[14] J. Flynn, S. Dance, and D. Schaefer, “Industry 4.0 and ts Potential Impact on Employment Demographics in the UK,” Oct. 2017. doi: 10.3233/978-1-61499-792-4-239.

[15] “Industry 5.0 - from virtual to physical | LinkedIn.” https://www.linkedin.com/pulse/industry-50-from-virtual-physical-michael-rada/ (accessed Oct. 07, 2021).

[16] “Industry 5.0 definition | by Michael Rada | Medium.” https://michael-rada.medium.com/industry-5-0-definition-6a2f9922dc48 (accessed Jul. 29, 2021).

[17] P. Skobelev and B. S. Yu., “On the way from Industry 4.0 to Industry 5.0: from digital manufacturing to digital society,” 2017. Accessed: Oct. 07, 2021. [Online]. Available: https://stumejournals.com/journals/i4/2017/6/307

[18] V. Kharchenko, O. Morozova, O. Illiashenko, and & S. Sokolov, “A Digital Twin for the Logistics System of a Manufacturing Enterprise Using Industrial IoT,” vol. 47, no. 1, pp. 125–134, 2020, doi: 10.11610/isij.4708.

[19] R. Boschert Stefan and Rosen, “Digital Twin—The Simulation Aspect,” in *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers*, D. Hehenberger Peter and Bradley, Ed. Cham: Springer International Publishing, 2016, pp. 59–74. doi: 10.1007/978-3-319-32156-1_5.

[20] M. Grieves, “Origins of the Digital Twin Concept,” *Florida Institute of Technology*, Aug. 2016, doi: 10.13140/RG.2.2.26367.61609.

[21] “Digital model, digital shadow, or digital twin – what is at the core of data-driven shipbuilding? / Blog / CADMATIC.” https://www.cadmatic.com/en/resources/blog/digital-model,-digital-shadow,-or-digital-twin-%E2%80%93-what-is-at-the-core-of-data-driven-shipbuilding/ (accessed Aug. 04, 2021).

[22] Gartner, “Gartner Survey Reveals Digital Twins Are Entering Mainstream Use.” https://www.gartner.com/en/newsroom/press-releases/2019-02-20-gartner-survey-reveals-digital-twins-are-entering-mai (accessed Aug. 04, 2021).

[23] E. Glaessgen and D. Stargel, *The digital twin paradigm for future NASA and U.S. air force vehicles*. 2012. doi: 10.2514/6.2012-1818.

[24] A. Fuller, Z. Fan, C. Day, and C. Barlow, “Digital Twin: Enabling Technologies, Challenges and Open Research,” *IEEE Access*, vol. 8, pp. 108952–108971, 2020, doi: 10.1109/ACCESS.2020.2998358.

[25] J. Vachálek, L. Bartalský, O. Rovný, D. Šišmišová, M. Morháč, and M. Lokšík, “The digital twin of an industrial production line within the industry 4.0 concept,” in *2017 21st International Conference on Process Control (PC)*, 2017, pp. 258–262. doi: 10.1109/PC.2017.7976223.

[26] S. Boschert, C. Heinrich, and R. Rosen, “Next Generation Digital Twin,” in *Proceedings of the 12th International Symposium on Toolsand Methods of Competitive Engineering, Las Palmas de Gran Canaria, Spain*, Nov. 2018, pp. 209–217.

[27] H. Elayan, M. Aloqaily, and M. Guizani, “Digital Twin for Intelligent Context-Aware IoT Healthcare Systems,” *IEEE Internet of Things Journal*, p. 1, 2021, doi: 10.1109/JIOT.2021.3051158.

[28] T. Ruohomäki, E. Airaksinen, P. Huuska, O. Kesäniemi, M. Martikka, and J. Suomisto, “Smart City Platform Enabling Digital Twin,” in *2018 International Conference on Intelligent Systems (IS)*, 2018, pp. 155–161. doi: 10.1109/IS.2018.8710517.

[29] M. R. Lopes, A. Costigliola, R. Pinto, S. Vieira, and J. M. C. Sousa, “Pharmaceutical quality control laboratory digital twin – A novel governance model for resource planning and scheduling,” *International Journal of Production Research*, vol. 58, no. 21, pp. 6553–6567, 2020, doi: 10.1080/00207543.2019.1683250.

[30] K. H. and S. G. and W. J. Błazewicz Jacek and Ecker, “Static Shop Scheduling,” in *Scheduling in Computer and Manufacturing Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 171–191. doi: 10.1007/978-3-642-79034-8_6.

[31] J. R. Jackson, “Simulation research on job shop production,” *Naval Research Logistics Quarterly*, vol. 4, no. 4, pp. 287–295, 1957, doi: https://doi.org/10.1002/nav.3800040404.

[32] A. Fahmy, T. Hassan, and H. Bassioni, “What is Dynamic Scheduling?,” *PM World Journal*, vol. 3, Aug. 2014.

[33] D. Ouelhadj and S. Petrovic, “A survey of dynamic scheduling in manufacturing systems,” *Journal of Scheduling 2008 12:4*, vol. 12, no. 4, pp. 417–431, Oct. 2008, doi: 10.1007/S10951-008-0090-8.

[34] K. Muhamadin, M. Shukri, and O. Khayal, “A Review for Dynamic Scheduling in Manufacturing,” *The Global Journal of Researches in Engineering*, vol. 18, no. 5-J, pp. 25–37, Aug. 2018, doi: 10.13140/RG.2.2.15345.33129.

[35] P. Sharma and A. Jain, “Analysis of dispatching rules in a stochastic dynamic job shop manufacturing system with sequence-dependent setup times,” *Frontiers of Mechanical Engineering*, vol. 9, no. 4, pp. 380–389, Dec. 2014, doi: 10.1007/S11465-014-0315-9.

[36] P. D. D. Dominic, S. Kaliyamoorthy, and M. S. Kumar, “Efficient dispatching rules for dynamic job shop scheduling,” *The International Journal of Advanced Manufacturing Technology 2004 24:1*, vol. 24, no. 1, pp. 70–75, May 2004, doi: 10.1007/S00170-002-1534-5.

[37] Averill M. Law and David Kelton, *Simulation Modeling and Analysis*, 5th ed. New York, NY, USA: McGraw-Hill, 2015.

[38] A. Alrowaie Ahmed, “The effect of time-advance mechanism in modeling and simulation,” Calhoun, M.S. thesis, Monterey, USA, 2011. Accessed: Aug. 22, 2021. [Online]. Available: https://calhoun.nps.edu/handle/10945/10798

[39] Jerry Banks, John Carson, Barry Nelson, and David Nicol, *Discrete-event System Simulation. 5th ed.* 2016.

[40] D. R. C. Hill, “Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems: Second Edition by B. P. Zeigler, H. Praehofer, T. G. Kim, Academic Press, San Diego, CA, 2000.,” *International Journal of Robust and Nonlinear Control*, vol. 12, no. 1, pp. 91–92, Jan. 2002, doi: 10.1002/RNC.610.

[41] H. Yu, S. Han, D. Yang, Z. Wang, and W. Feng, “Job Shop Scheduling Based on Digital Twin Technology: A Survey and an Intelligent Platform,” *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/8823273.

[42] W. Hofmann and F. Branding, “Implementation of an IoT- and Cloud-based Digital Twin for Real-Time Decision Support in Port Operations,” *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2104–2109, Jan. 2019, doi: 10.1016/J.IFACOL.2019.11.516.

[43] P. Karagiannis, N. C. Zacharaki, G. Michalos, and S. Makris, “Increasing flexibility in consumer goods industry with the help of robotized systems,” *Procedia CIRP*, vol. 86, pp. 192–197, Jan. 2019, doi: 10.1016/J.PROCIR.2020.01.039.

[44] T. Allweyer, *BPMN 2.0*. Norderstedt, Germany: BoD, 2010.

[45] P. Harmon, “The State of Business Process Management 2016,” 2016, Accessed: Aug. 10, 2021. [Online]. Available: https://www.bptrends.com/bpt/wp-content/uploads/2015-BPT-Survey-Report.pdf

[46] “BPMN - ML Wiki.” http://mlwiki.org/index.php/BPMN (accessed Dec. 16, 2021).

[47] Anylogic, “Help Section.” https://anylogic.help/ (accessed Sep. 09, 2021).

[48] “Preparing to Measure Process Work with a Time Study.” https://www.isixsigma.com/methodology/business-process-management-bpm/preparing-measure-process-work-time-study/ (accessed Aug. 25, 2021).

[49] Institute of Industrial & Systems Engineers, “Z94.17 Work Design and Measurement.” https://www.iise.org/Details.aspx?id=2606 (accessed Aug. 25, 2021).

[50] N. Magagnotti and R. Spinelli, “Good practice guidelines for biomass production studies,” CNR Ivalsa, 2012.

[51] ABB, “Technical data for the ABB IRB 1200 industrial robot | ABB Robotics - IRB 1200 | ABB Robotics (Industrial Robots | ABB Robotics).” https://new.abb.com/products/robotics/industrial-robots/irb-1200/irb-1200-data (accessed Aug. 27, 2021).

[52] Anylogic, “Experiment replication | anyLogistix Help.” https://anylogistix.help/experiments/replications.html (accessed Sep. 16, 2021).

[53] P. Thompson and Y. Liu, “UNDERSTANDINGS OF MARGIN OF ERROR,” Sep. 2005, doi: 10.21037/jtd.2017.09.14.

Appendix A

Complete Resource Allocation Study

A.1 Parallel Tasks

The prototype currently only supports single tasking, where a new job is only processed when the last one is completed, as said in the introduction, one of the goals of this thesis is to see the effect of parallel tasking in the completion time, with different heuristics. For this, an iteration with single tasks was compared with the parallel tasking with the dispatching rules referred in Chapter 3s Scheduling and Algorithm, with 50 bottles:

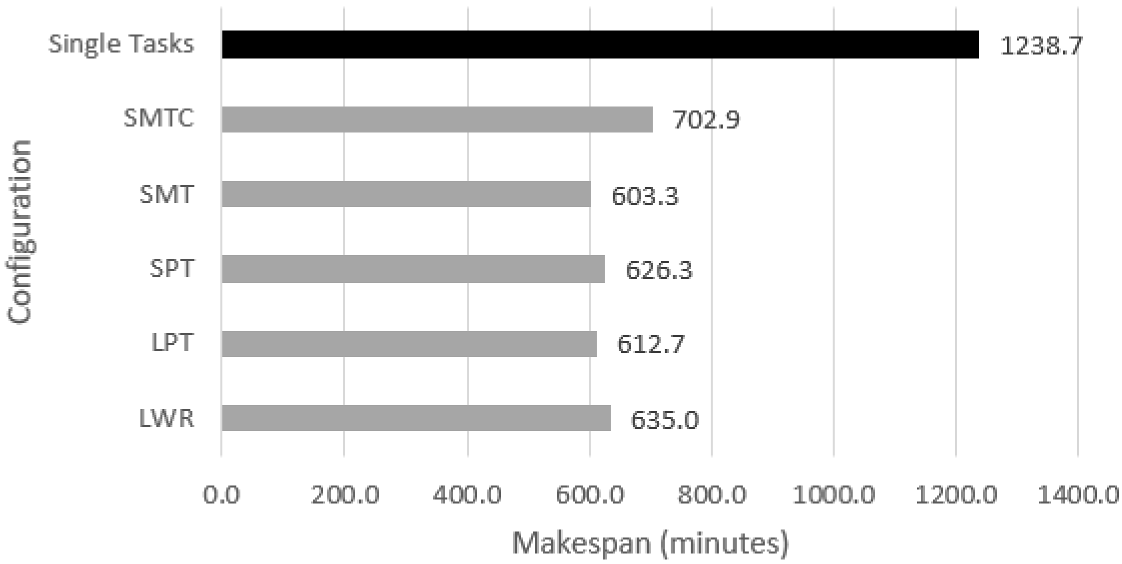

Figure A. 1: Makespan comparison of single tasking with parallel tasking with the best performing SMT rule

Resource	Utilization (%)
Robot	23.9
W1	3.8
W2	7.7
W3	36.3
W4	2.1
W5	1.9
W6	20.0
W7	3.4
W8	0.9

Table A. 1: Utilization of resources using single tasking

	CSMT	SMT	SPT	LPT	LWR
Performance Improvement (%)	43.3	51.3	49.4	50.5	48.7

Table A. 2: Performance improvement of utilizing parallel tasks, compared to single tasking

	CSMT		SMT		SPT		LPT		LWR	
Resource Utilization/Occupation	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w
Robot	45.9		60.6		58.6		60.6		58.0	
W1	6.8	47.4	7.8	45.0	7.6	44.9	7.8	46.2	7.5	44.2
W2	13.6	72.2	15.7	78.1	15.3	77.5	15.6	68.4	15.1	67.9
W3	64.5	83.2	74.3	82.1	72.4	79.7	74.1	83.7	71.9	79.8
W4	3.7	6.5	4.3	20.1	4.2	22.9	4.3	21.0	4.2	20.5
W5	3.3	31.4	3.8	20.5	3.7	12.2	3.8	18.3	3.7	11.0
W6	35.6	44.6	41.1	49.1	40.0	47.2	41.0	48.3	39.5	46.2
W7	6.1	22.0	7.0	22.2	6.8	20.9	7.0	22.3	6.7	19.2
W8	1.5	3.2	1.8	5.0	1.7	4.5	1.8	5.3	1.7	4.6

Table A. 3: Resource utilization and occupation of the parallel tasks configuration

Combining parallel tasking with the SMT dispatching rule originates the best results, reducing the overall completion time by 51.3%, when comparing with single tasking, making this change imperative to improve the system, as well as allowing the system to be improved by adding identical parallel machines in key processes. The utilization of the robot increased significatively, from 23.3% to 60.6% when utilizing the SMT dispatching rule, all workstations' utilization also increased dramatically, which resulted into the significant reduction in makespan.

A.2 Home Position

Individuals involved in the development of the prototype claimed the robot goes to the home position for safety reasons, as it's a prototype, but they claim that in the future its intended for the need to go to the home position in not necessary, and so, the model was put to test to determine the effect of the home position in the performance, the model was already built for this possibility and it was only necessary to run the simulation with the chosen number of replications.

Different scheduling policies were also applied to find the most fitting for the the following results were obtained for different dispatching rules:

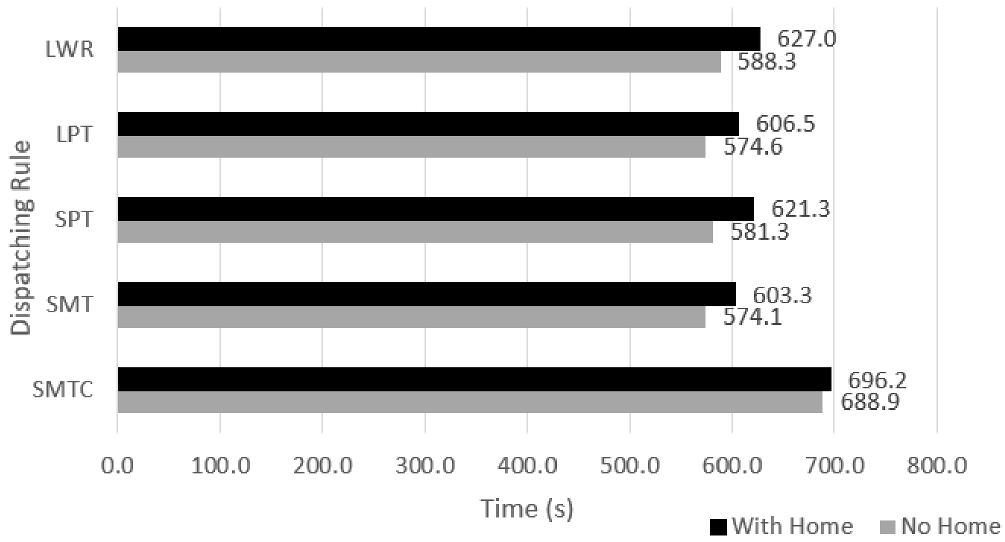

Figure A. 2: Makespan comparison of going to the home position across all dispatching rules

	CSMT	SMT	SPT	LPT	LWR
Performance Improvement (%)	1.0	4.8	6.4	5.3	6.2

Table A. 4: Performance improvement of not going to the home position

As seen on figure A.1 and table A.1, going to the home position does not affect the process time dramatically, this is because the rotation time of the robot is small in comparison to the processing times.

A.3 Resource Allocation

To better understand where the model can be improved, various experiments with different parallel machine configurations were ran for the different dispatching rules, the table A.2, was developed with information relative to workstation and robot utilization and occupation for the model with single machines, and no home position:

	CSMT		SMT		SPT		LPT		LWR	
Resource Utilization/Occupation	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w
Robot	39.8		51.1		50.3		51.4		50	
W1	6.9	43.1	8.3	40.8	8.2	41.5	8.3	42.7	8.1	41.5
W2	13.9	75.9	16.7	82	16.5	80	16.7	72.1	16.3	71.6
W3	66.4	85.3	78.8	85.8	77.8	84.6	78.9	86.7	77.4	84.4
W4	2.6	6	3.1	20.2	3	22.2	3.1	20.7	3	19.9
W5	3.4	31.6	4.1	16.1	4	12	4.1	15.1	4	10.8
W6	36.4	44.6	43.7	50.7	43.1	49.6	43.7	50.2	42.6	48.8
W7	6.2	25	7.4	17.5	7.3	17.7	7.4	18.6	7.3	16.8
W8	1.6	3.3	1.9	4.2	1.9	4.1	1.9	4.4	1.8	4

Table A. 5: Resource utilization and occupation with no home position with all dispatching rules

After testing different scheduling policies, and analyzing tables A.1 and A.2 as well as figure A.1, and recurring to the visualization, it was shown that apart from the CSMT, the dispatching rules didn't have a large effect on the utilization of resources and makespan, this is due to the fact that, most of the time, no more than one bottle is available, this is due to the necessary availability conditions restricting the possible movements, and the large gap between short and long processing times, which creates bottlenecks, by the utilization gap between Ws, of long processes (W3, W6) and short ones (W1,W4, W7, W8).

The CSMT was not very successful, as waiting for a long process to finish, while other are available, is most of the times, not a good decision, CSMT might be better applied for situations with short processing times relative to the robot speed.

The W3 seems to have a largest utilization and occupation rate, and its predecessors in the workflow, shown in figure 3.7, W2 and W4, have a large difference between its occupation and utilization rates, implying that the bottles at these Ws spend most of the time waiting, indicating a possible bottleneck at W3. The blocked W2, also affects W1, although, it is also due to it being necessary for each job to be processed 3 times there, and so, there is constant need to use this workstation.

The robot has a low utilization rate, as it is idle almost half the time, with the right resource allocation, it is expected to increase its utilization.

As W3 is a possible bottleneck, its resource pool capacity was increased by one in the model, in the real world this mean having a parallel W3 workstation, the model was then run again with this change:

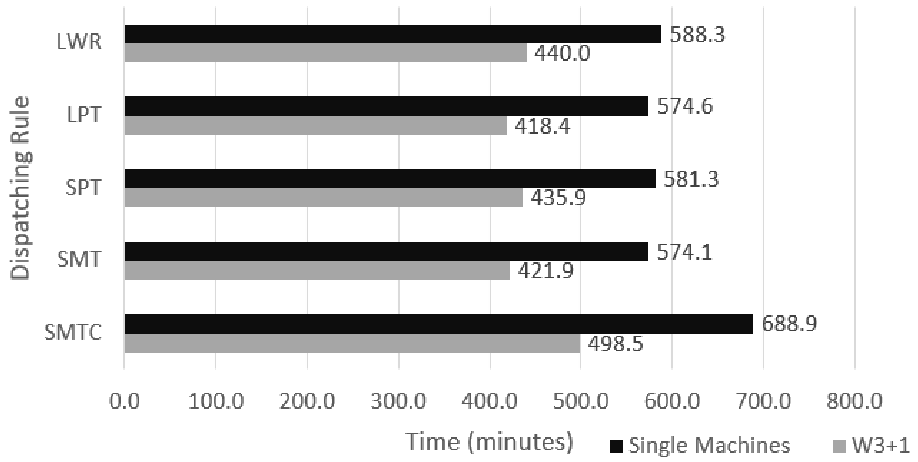

Figure A. 3: Makespan comparisson of adding one W3 across all dispatching rules

Performance Improvement (%)	**SMTC**	**SMT**	**SPT**	**LPT**	**LWR**	**Average**
Single Machines to W3+1	27.6	26.5	25.0	27.2	25.2	26.0

Table A. 6: Performance improvement of adding one W3 across all dispatching rules

	CSMT		**SMT**		**SPT**		**LPT**		**LWR**	
Resource Utilization/Occupation	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w
Robot	59.5		71.7		69.3		72.6		69	
W1	9.5	51.4	11.3	61.1	10.9	56.6	11.4	57.3	10.8	54.4

W2	19.2	38.1	22.6	49.3	21.9	49.9	22.8	44	21.7	45.8
W3	45.4	63.5	53.6	65.7	52	61.6	54.2	68.9	51.6	61.8
W4	3.6	13.3	4.2	19.8	4.1	20	4.3	19.4	4.1	18.3
W5	4.7	60.6	5.5	53.9	5.4	31.2	5.6	48.1	5.3	27.9
W6	50	69.9	59.2	70.3	57.3	67	59.6	69.8	56.8	66.1
W7	8.5	33.6	10.1	27.9	9.8	28.9	10.2	31	9.7	27.6
W8	2.2	5.1	2.6	6.7	2.5	5.9	2.6	7.8	2.5	5.8

Table A. 7: Utilization and occupation when adding one W3 across all dispatching rules

As seen on table X, adding a parallel W3 yields an increase in performance of about 26% across all dispatching rules, proving the W3 is a bottleneck, applying this change in the real counterpart proves effective independently of the algorithm chosen.

With this new configuration, the W3 might still be a bottleneck as its utilization/occupation ratio is still high, W6 might also be another possible bottleneck, as it has both a large utilization and occupation, and its predecessors in the workflow W1 and W5, have a very low utilization rate, compared to the occupation, so the bottles are waiting for the W6 to be free, figure X displays the makespan with the two parallel W3 and W6:

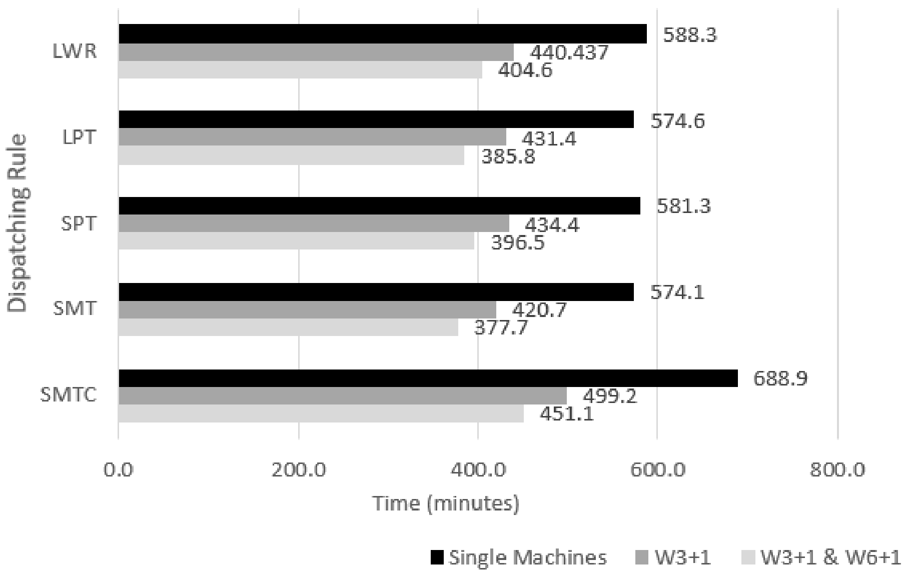

Figure A. 4: Makespan comparison between single machines, one extra W3, and adding one W3 and W6 across all dispatching rules

To better understand where resources should be allocated, with the extra W3 and W6, tables X and X were created:

Performance Improvement (%)	SMTC	SMT	SPT	LPT	LWR	Average
W3+1 to W3+1 & W6+1	9.6	10.2	8.7	10.6	8.1	9.5

Table A. 8: Performance improvement of adding one W3 and W6, compared with adding one W3 across all dispatching rules

	CSMT	SMT	SPT	LPT	LWR

Resource Utilization/Occupation	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w
Robot	65		79.3		76		77.8		74.1	
W1	10.6	46.5	12.6	50.3	12	46.5	12.3	48.5	11.8	43.9
W2	21.2	51.6	25.2	63.8	24.1	65.3	24.8	54.6	23.6	56.1
W3	50.4	70.1	60.2	74.5	57.2	68.3	58.6	76.5	55.9	68.4
W4	4	15.8	4.7	24.7	4.5	26.1	4.6	24.2	4.4	22.8
W5	5.2	54.4	6.2	46.3	5.9	21.8	6.1	44.6	5.8	20.7
W6	27.7	43.6	33.1	42.9	31.6	40.7	32.4	40.8	30.9	37.7
W7	9.5	35.3	11.3	32.8	10.7	34.1	11	36	10.5	28.6
W8	2.4	6.1	2.9	7.7	2.7	6.9	2.8	9.6	2.7	6.8

Table A. 9: Utilization and occupation of adding one W3 and W6 across all dispatching rules

According to the simulation model adding an extra machine to W3 and W6, further decreases makespan by about 32.9%, since the existing W6 already allows two containers simultaneously, there is the possibility of only adding the W3, and using the extra space in W6.

W3 still possesses the highest utilization, so it still restricting the workflow and W6 has still the 2nd overall highest utilization therefore, increasing the capacity of these resource pools should be experimented. As W4 has W3 as its predecessor in the workflow, adding an extra W4 can also be tested, as it might reduce the W3 occupation.

W1 appears to restrict the flow of tasks, as W5 has a low utilization/occupation ratio, since each job need to go through W1, having a single machine might create a choke point, and so having an extra parallel W1 might prove beneficial.

With this information, combinations of adding extra W1, W3 and W6 were experimented, to compare diferent alternatives to adding parallel W3 & W1 with the following results:

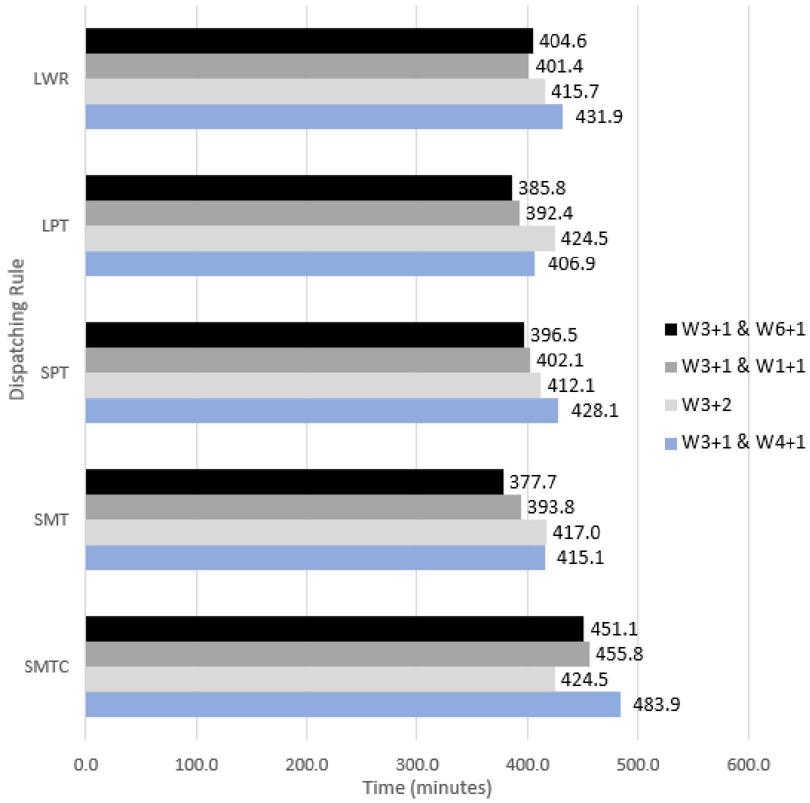

Table A. 10: Makespan comparison relative to adding one W3 and W6, across all dispatching rules

A table of performance increase relative to the W3+1 configuration can be seen bellow:

Performance Improvement (%)	SMTC	SMT	SPT	LPT	LWR	Average
W3+1 to W3+1 & W6+1	9.6	10.2	8.7	10.6	8.1	9.5
W3+1 to W3+1 & W1+1	8.7	6.4	7.5	9	8.9	8.1
W3+1 to W3+2	15	0.9	5.1	1.6	5.6	5.6
W3+1 to W3+1 & W4+1	3.1	1.3	1.4	5.7	1.9	2.7

Table A. 11: Makespan reduction relative to adding one W3, across all dispatching rules

Adding only two parallel W3 does not decrease makespan by a large margin, overshadowed by adding a parallel machine to W6 or W1, except when using the SMTC dispatching rule, which is the only case so far, where SMTC can be considered.

Between adding a parallel W1, W6 or W4, adding a W6 seems to be slightly better in reducing makespan, and as said before, its possible to have two W6 more easily than any other Ws.

As the combination of extra parallel W3 and W6 has the lowest makespan for most dispatching rules, so the next experiments in resource allocations are based on the results and comments from figure A.3 and table A.6.

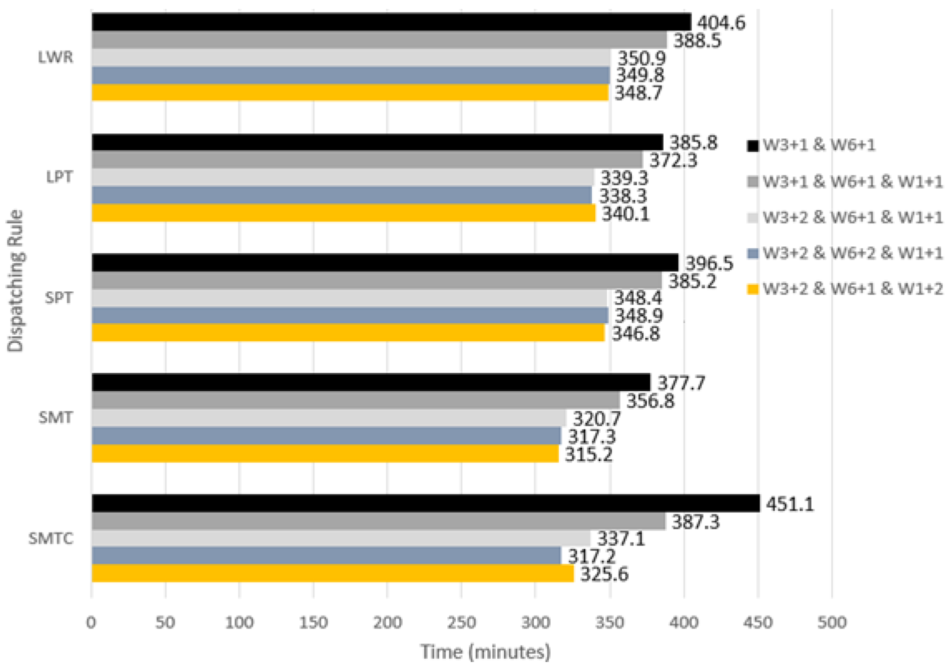

Figure A. 5: Makespan comparisson based off adding one W3 and W6

A table of performance increase relative to the W3+1 & W6+1 configuration can be seen bellow:

Performance Improvement (%)	**SMTC**	**SMT**	**SPT**	**LPT**	**LWR**	**Average**
W3+1 & W6+1 & W1+1	14.1	5.5	2.9	3.5	4	6
W3+2 & W6+1 & W1+1	25.3	15.1	12.1	12.1	13.3	15.6
W3+2 & W6+2 & W1+1	29.7	16	12	12.3	13.5	16.7
W3+2 & W6+1 & W1+2	27.7	16.4	12.4	11.9	13.9	16.5

Table A. 12: Makespan reduction relative to adding one W3 and W6

Adding an extra W1 reduces the makespan by about 6%, combined with another W3, further reduces it by 15.6%, adding others only renders a reduction of less than 2%, therefore, adding more parallel machines, does not decrease makespan significatively, and so, no more configurations were experimented, table A.10 shows the utilization/occupation of this configurations:

	CSMT		**SMT**		**SPT**		**LPT**		**LWR**	
Resource Utilization/Occupation	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w	U_w	O_w
Robot	87.7		92.6		87.6		89.7		86.4	
W1	7.1	71.7	7.4	71.4	6.8	66.2	7	65.1	6.8	55.7
W2	28.4	60.7	29.8	61.4	27.4	72.9	28.2	57	27.2	63.2
W3	45.2	65.4	47.1	66	43.4	56.8	44.7	74.6	43.3	60.2
W4	5.4	30.4	5.6	35.1	5.1	39.4	5.3	38.7	5.1	31.4
W5	6.9	63.9	7.3	62.4	6.7	26.6	6.9	53.6	6.7	24.9
W6	37.1	62.1	38.9	61	35.8	49.2	36.8	48.9	35.6	44.1
W7	12.6	38.1	13.3	37.7	12.2	39.7	12.5	43.7	12.1	32.4
W8	3.2	8.5	3.4	9.3	3.1	8.5	3.2	11.4	3.1	8.3

Table A. 13: Utilization and occupation table of adding two W3 and one W1 and W6

The robot is now utilized most of the time with its high utilization, the movement time might significatively be hindering the workflow, as the workstations still have an overall low utilization, and increasing the number of parallel machines no longer yields reductions in makespan larger than 2%, also the fact that the SMT rule, which relates with the robot's speed, is performing better than other dispatching rules, makes a study on the robot movement speed a relevant topic.

Appendix B

Samples of Customer Requests

Recipes	W1	W2	W3	W4 1	W4 2	W5	W1 2	W6	W7	W1 3	W8
1	14.5	58.4	377.1	10.0	25.0	14.0	6.5	150.0	25.5	7.5	0.0
2	14.5	69.6	298.3	0.0	0.0	14.0	6.5	150.0	25.5	7.5	27.0
3	14.5	64.7	318.9	0.0	0.0	14.0	6.5	150.0	25.5	7.5	0.0
4	14.5	58.7	272.6	10.0	25.0	14.0	6.5	150.0	25.5	7.5	27.0
5	14.5	62.0	286.3	0.0	0.0	14.0	6.5	150.0	25.5	7.5	27.0
6	14.5	61.2	301.7	10.0	25.0	14.0	6.5	150.0	25.5	7.5	0.0
7	14.5	53.2	351.4	10.0	25.0	14.0	6.5	150.0	25.5	7.5	0.0
8	14.5	60.6	288.0	0.0	0.0	14.0	6.5	150.0	25.5	7.5	27.0
9	14.5	56.2	356.6	0.0	0.0	14.0	6.5	150.0	25.5	7.5	0.0
10	14.5	68.8	378.9	0.0	0.0	14.0	6.5	150.0	25.5	7.5	0.0
11	14.5	68.8	334.3	0.0	0.0	14.0	6.5	150.0	25.5	7.5	0.0
12	14.5	68.0	327.4	10.0	25.0	14.0	6.5	150.0	25.5	7.5	0.0
13	14.5	74.0	342.9	10.0	25.0	14.0	6.5	150.0	25.5	7.5	0.0
14	14.5	60.6	282.9	0.0	0.0	14.0	6.5	150.0	25.5	7.5	0.0
15	14.5	66.9	293.1	10.0	25.0	14.0	6.5	150.0	25.5	7.5	27.0
16	14.5	51.6	305.1	10.0	25.0	14.0	6.5	150.0	25.5	7.5	0.0
17	14.5	72.4	270.9	10.0	25.0	14.0	6.5	150.0	25.5	7.5	0.0
18	14.5	59.0	334.3	0.0	0.0	14.0	6.5	150.0	25.5	7.5	27.0
19	14.5	64.2	365.1	0.0	0.0	14.0	6.5	150.0	25.5	7.5	0.0
20	29.0	139.3	298.3	0.0	25.0	28.0	13.0	300.0	51.0	15.0	27.0
21	29.0	94.8	365.1	10.0	0.0	28.0	13.0	300.0	51.0	15.0	27.0
22	29.0	74.6	282.9	0.0	25.0	28.0	13.0	300.0	51.0	15.0	27.0
23	29.0	85.5	270.9	10.0	0.0	28.0	13.0	300.0	51.0	15.0	0.0
24	29.0	76.2	363.4	0.0	25.0	28.0	13.0	300.0	51.0	15.0	27.0
25	29.0	81.1	281.1	10.0	0.0	28.0	13.0	300.0	51.0	15.0	27.0
26	29.0	70.8	318.9	0.0	25.0	28.0	13.0	300.0	51.0	15.0	27.0
27	29.0	86.0	262.3	10.0	0.0	28.0	13.0	300.0	51.0	15.0	27.0
28	29.0	87.7	313.7	0.0	25.0	28.0	13.0	300.0	51.0	15.0	0.0
29	29.0	98.6	269.1	10.0	0.0	28.0	13.0	300.0	51.0	15.0	27.0
30	29.0	76.8	337.7	10.0	25.0	28.0	13.0	300.0	51.0	15.0	27.0

Table B. 1: Sample of 30 customer recipes, from a list containing 500 recipes