POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria della produzione industriale e dell'innovazione tecnologica

Tesi di Laurea Magistrale

Software simulation of the performance of an automated warehouse with material handling through mini-load stacker crane and AGVs systems

Relatore

Carlo Rafele

Correlatore

Giovanni Zenezini

Candidato

Andrea Ferrari

Ottobre 2020

Ringraziamenti

Giunto al termine del mio percorso di formazione e prima di iniziare a descrivere il mio lavoro di tesi, vorrei dedicare alcune righe a tutti coloro che mi hanno aiutato e sostenuto in questo viaggio che influenzerà per sempre la mia vita.

In primo luogo, vorrei ringraziare il professore Rafele, relatore di questa tesi, e il mio correlatore Zenezini, per l'opportunità offertami con questo progetto e per la disponibilità e supporto dimostratemi durante il periodo di ricerca e nella fase di stesura di questo elaborato. Un ringraziamento generale va anche al Politecnico di Torino, che mi ha concesso la possibilità di approfondire la mia conoscenza riguardo le tematiche affrontate durante il mio percorso di studi.

Un caloroso abbraccio va a mia mamma Emanuela e mia sorella Sara, che, con il loro sostegno morale ed economico, hanno contribuito a rendermi una persona migliore e hanno permesso che io raggiungessi questo importante traguardo. Mando anche un profondo pensiero a mio padre Maurizio, il quale mi ha sempre insegnato a cercare di fare la differenza in tutti i modi e impegnarmi a fondo anche nelle piccole cose e che ora da lassù veglia su di me e mi dà la forza per andare avanti. Un grazie va anche rivolto ai miei nonni Angelo e Maria, a mio zio Giovanni e ai miei cugini Giulia, Letizia e Francesco, che sono riusciti a strapparmi un sorriso anche nei momenti più difficili.

Doveroso è anche ringraziare la mia ragazza e compagna di corso Azzurra, che mi ha sostenuto e sopportato durante le sessioni interminabili e intense di studio, che mi ha convinto a non arrendermi quando la speranza iniziava a svanire, che mi è stata sempre vicina e che ha condiviso con me la maggior parte del periodo di lavoro su questa tesi.

Un pensiero è rivolto anche a tutti i miei colleghi di università con i quali ho vissuto momenti felici e tristi, sia al Politecnico di Torino, sia durante due importanti esperienze di Erasmus in Spagna e Irlanda. Il fantastico rapporto che si è creato nel corso degli anni è stato fondamentale per arrivare qui dove sono ora.

In conclusione, ringrazio anche tutti i miei amici di Castagnito, paese dal quale provengo, per essermi stati di supporto lungo tutto il percorso universitario e per avermi reso più spensierati circostanze e momenti complicati.

Un sentito grazie a tutti.

Table of contents

Table of figures

Table of tables

Introduction

Simulation modelling is a business procedure that is becoming increasingly widely used to develop processes. With the advent of up-to-grade technologies, the concepts of simulation modelling find easy application to business processes improvement.

The object of this research is to show an example of how the simulation modelling theories and principles may be applied to a case study. In particular, the system into analysis consists of an automated warehouse characterized by handling of materials though innovative technologies like a mini-load stacker crane and automated guided vehicles. It consists of a project proposed by Polytechnic of Turin, that planned to build the warehouse in the future for direct research purposes. The objective of this study is the development of a virtual model of this warehouse in the simulation software AnyLogic, in order to study its performance prior to its physical implementation. Constructing some experiments, it is possible to observe the behavior of the system in different conditions and establish the optimal design solution to be implemented and where reside the major criticalities.

The research methodology consists of different steps. Initially, there was a phase of literature study which regards the general principles of supply chain management, the concepts of digitalization of processes and the theories underlying simulation modelling. With this basic knowledge, the research has continued with the analysis of the case study in order to identify which are the fundamental elements to consider when building the model. This phase was followed by the construction of a conceptual model, which consists of the description of all the elements that characterized the model, including all the assumption and simplification of the abstraction process. Having built the conceptual model, it was translated into AnyLogic language. In order to do that, a deep study of the software has been performed and existing models were examined to draw inspiration from them. The construction of the virtual model was then followed by the verification process and the statistical analysis of the result obtained from simulations.

The thesis is composed of 7 chapters divided as follows. The first chapter illustrates the concept of supply chain, how a supply chain is formed, and which are the main actors involved. Additionally, it enunciates some theories and principles regarding supply chain management and its importance in terms of business success. The second chapter contains an analysis of what is the industry 4.0 and its impacts on the supply chain. This chapter also introduces the idea of standardization of processes and it analyses the principal technologies applied in warehouses. Finally, the concept of digital twin is illustrated. The third chapter presents the major theories regarding simulation, simulation modelling, and conceptual modelling. It also describes the main simulation modelling paradigms. The fourth chapter illustrates the case study and the project specifications. The fifth chapter contains a detailed presentation of the conceptual model, with all the elements included in the model, the assumptions made and the input/output of it. The sixth chapter describes the main features of the simulation software AnyLogic and how the conceptual model is translated into computer code. The final chapter illustrates which are the experiments conducted on the model and the results obtained from simulations.

Thanks to this research, it was achievable to find an optimal design solution, to understand which are the major criticalities of the warehouse into analysis and which are the aspects that can be improved.

1. Supply Chain Management and efficiency in processes

1.1 Introduction to supply chain

1.1.1 The concept of supply chain

"A supply chain consists of all the stages involved, directly or indirectly, in fulfilling a customer request. The supply chain not only includes the manufacturer and suppliers, but also transporters, warehouses, retailers, and customers themselves" (Chopra and Meindel, 2007). In other words, a supply chain can be considered as the network of organizations, individuals, resources and activities involved in the entire process of creating and selling a product, including every single stage from the supply of raw materials and the manufacture of goods to their distribution and sell to the end user or client.

Generally, it is possible to identify three main phases that characterize a supply chain, which are further divided in minor processes. These are:

- Supply, that refers to all the aspects concerning the processes of raw materials request necessary to the production of products;
- Production, which consists of the manufacturing activities that employ raw materials to generate final goods;
- Distribution, that is composed by all the operations that allow a product to be delivered to the customer. This aspect of supply chain can be viewed as the result of the action of warehouses, wholesaler and retailers.

A supply chain network is characterized by the presence of three different flows within and among companies, that are:

- Products flow;
- Information flow;
- Finances flow.

Products or physical flow involves the movement of goods from supplier to consumer. The consumer can be external if it consists of another company or the final consumer, or it can be internal if it represents a subsequent process of the same company. Generally, in a supply chain materials and goods flows downstream, from supplier to consumer. There may be also a reverse flow of products, principally associated with rejections or goods returns (Bsaikrishna, 2016).

On the other hand, information flow permits the various subjects that compose the supply chain to coordinate their long-term plans and to monitor the daily flows of materials within the network (Handfield, 2020). Types of information that flow along the supply chain are bills of materials, product data, descriptions and pricing, inventory levels, customer and order information, delivery scheduling, supplier and distributor information, delivery status, commercial documents, title of goods, current cash flow and financial information etc. (Bsaikrishna, 2016). Like product flow, information flow is bidirectional.

Finally, financial flow represents the movement of money from the customer to the supplier. Same as the other flows, financial flow may be bidirectional. Generally, once the customer receives the good and verifies it, the financial equivalent is payed, and money moves back to the supplier. In other cases, the financial flow may have the opposite direction, that is from supplier to customer, in a form of debit.

Figure 1.1 summarizes the general structure of a supply chain and the various flows within the nodes.

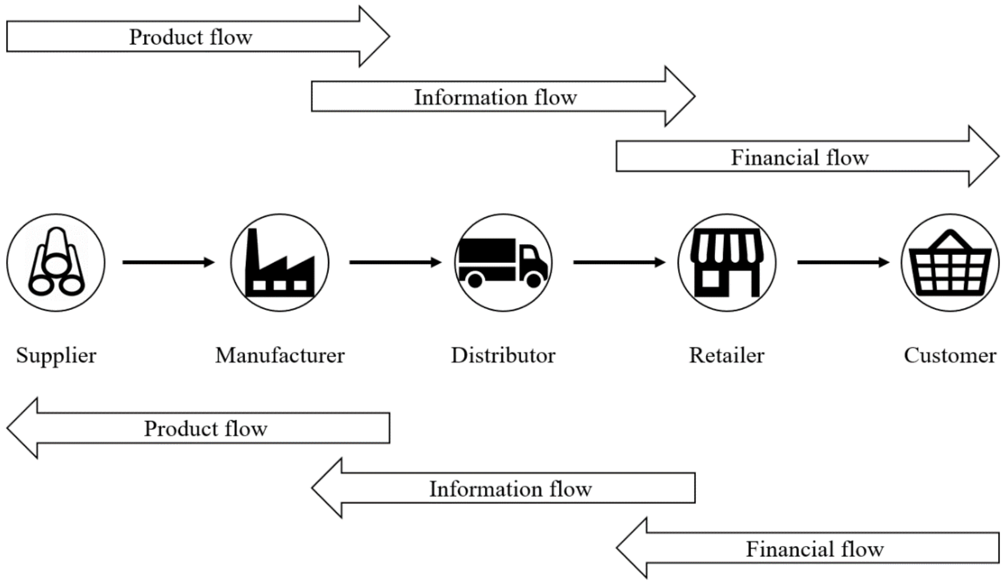


Figure 1.1 - Supply Chain Structure

1.1.2 The value chain

The concept of supply chain is often sustained with the idea of value chain. The notion of value chain was introduced in 1985 by the American academic Michael Ported in his book "Competitive Advantage: Creating and Sustaining Superior Performance." He presented the idea of value chain to explain how a business adds value to its raw material, in order to produce products that might be sold to customers.

Porter identified five steps in the value chain process:

- Inbound Logistics;
- Operations;
- Outbound Logistics;
- Marketing and sales;
- Service.

Inbound logistics consists of the processes that allow the reception and control of raw material, such as receiving, warehousing and inventory control. Operations are the activities that transform raw materials in final products. They differ in relation to the company that is creating value and may be assembly or manufacturing activities. Outbound logistics is composed by the activities that permit to move the final product to the consumer. These activities comprise warehousing, inventory management, order fulfilment and shipping. Marketing and sales incorporate all the activities that get a consumer to buy a specific product. Finally, service is composed by the activities that help to maintain and increase the value of goods, such as customer and warranty services. In addition to the five main steps, Porter identified in procurement, technology development, human resources management and infrastructure a series of support activities necessary to optimize the value chain performance. Value chain permits to a company to gain competitive advantage over competitors within the market, improving the activities in one of the steps above mentioned.

From this definition of what is a value chain and how it is structured, it is possible to recognize a finer shade of meaning from the idea of supply chain. The value chain consists of the process of adding value to raw materials to produce final goods. It gives business competitive advantage in the market. On the other hand, the supply chain represents all the processes that permit to get the product from the supplier to the customer. A performing supply chain leads to overall customer satisfaction.

1.1.3 B2B and B2C

Each supply chain has its roots in the relationships between the organizations that constitute the network. Two types of relationships exist: business-to-business (B2B) and business-to-consumer (B2C).

B2B refers to all those business transactions between businesses. Generally, the term is associated with the e-commerce world, but it may also refer to all the transactions completed in an industrial value chain, prior the sale to the final customer. Consequently, business-to-business indicates the kind of relationships that a company holds with its suppliers for procurement, production planning and monitoring activities, support in product development activities, or the relations that the enterprise has with its professional customers, that is other enterprises located further downstream in the supply chain.

On the other hand, B2C indicates the transactions between a company and the final consumer. These transactions take place through e-commerce platforms, stores or retailers. Business-to-consumer denotes the type of relationships that an enterprise has with its consumers or end users.

The key differences between B2B and B2C resides in volume of transactions, the number of clients, the length of supply chain, and negotiations strategies (Chan, 2019). The volume of B2B transactions is much higher than that of B2C transactions, since B2B sales are directed to business clients that require services or stock for manufacturing purposes or for resale. Despite this, B2B supply chains are characterized by a lower number of customers than B2C supply chains. Then, B2B supply chains are usually shorter than B2C. B2B supply chains involve a small number of actors concerned in the transaction. On the other hand, B2C supply chains include many producers, wholesalers and retailers prior the sale of the product or service to the final customer. In B2C supply chains, there is a lower level of negotiation between the company and the consumer and consequently the enterprise holds greater bargaining power. Alternatively, in B2B supply chains, the level of negotiation is larger among both companies.

1.2 Supply Chain Management

Supply Chain Management (SCM) consists of the operations that involve coordination and integration of materials, information and financial flows within and among companies that constitute a supply chain. "SCM deals with total business process excellence and represents a new way of managing the business and relationships with other members of the supply chain" (Cooper and Lambert, 2000). A correct management is achieved by operations like product development, sourcing, production, procurement, logistics. The great issue of Supply Chain Management is to optimize the processes in every element that composes the supply chain, from manufacturing sites and warehouses to inventory management, transportation and order fulfilments. Efficiency in a supply chain results in much lower costs and a faster production cycle.

A conceptual framework highlights the nature of Supply Chain Management (Lambert *et al.*, 1998). It consists of three closely inter-related elements: the supply chain network structure, the supply chain business processes, and the supply chain management components (Figure 1.2).

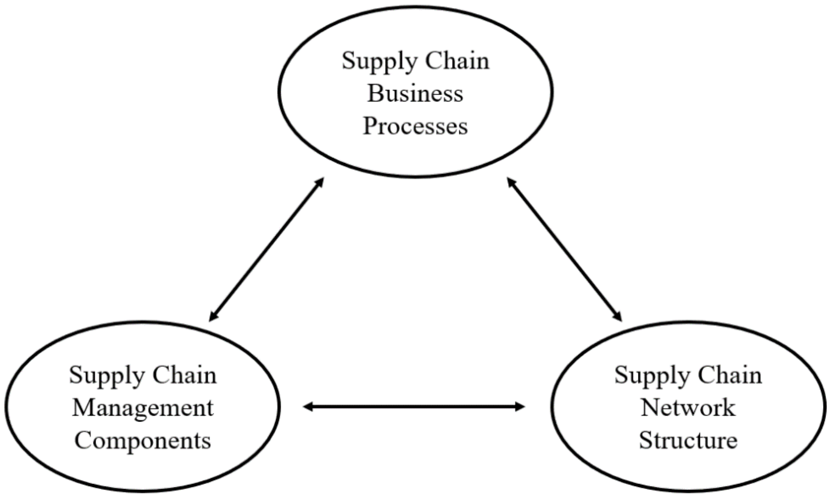


Figure 1.2 - SCM Framework (Adapted from: Lambert *et al.*, 1998)

1.2.1 Supply Chain Network Structure

Supply chains are networks of businesses that include all the stages from the supply of raw materials to the final consumer (see Figure 1.3). The performance of Supply Chain Management depends on several factors of the structure of the supply chain, that is the complexity of product, the number of available suppliers, and the availability of raw materials (Cooper and Lambert, 2000). Furthermore, other aspects to take into consideration in SCM comprise the length of the supply chain and the number of providers and buyers at each level of the network. Every point of the network requires different kind of relationship. It is fundamental to understand the proper closeness of partnership for each of these levels. Firm's capabilities and the importance of a company determine which point of the supply chain network needs greater attention.

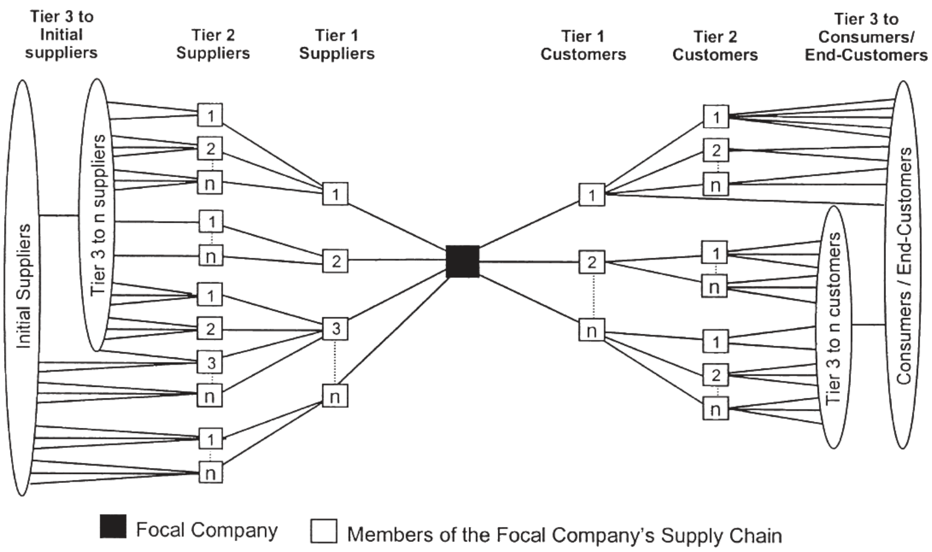


Figure 1.3 - Supply Chain Network Structure (Source: Lambert et al., 1998)

For this reason, it is fundamental to deeply understand the configuration of the supply chain network, in terms of the three main aspects that characterize a company's network structure (Cooper and Lambert, 2000):

- the members of the supply chain;
- the structural dimensions of the network;
- the different types of process links across the supply chain.

Identifying which are the members of the supply chain is helpful to determine the structure of the network. In particular, it is fundamental to recognize which are the critical nodes of the network, in order to guarantee the success of the business and efficiency in the supply chain. This step is central because of the high level of complexity that may generate if all the types of members are comprised. Indeed, "the members of a supply chain include all companies/organizations with whom the focal company interacts directly or indirectly through its suppliers or customers, from point of origin to point of consumption" (Cooper and Lambert, 2000). These members are classifiable in primary and supporting members. Primary members are "all those autonomous companies or strategic business units who carry out value-adding activities (operational and/ or managerial) in the business processes designed to produce a specific output for a particular customer or market". Supporting members "are companies that simply provide resources, knowledge, utilities, or assets for the primary members of the supply chain" (Cooper and Lambert, 2000). These two descriptions outline two important aspects of a supply chain network, that is the point of origin and the point of consumption. In fact, the point of origin is present where no preceding primary supplier exists, while the point of consumption occurs when the good or service is consumed by the customer and no additional value is added to it (Cooper and Lambert, 2000).

A supply chain network is describable though the analysis of its structural dimension. Three are three structural dimensions to take into considerations (Cooper and Lambert, 2000):

- the horizontal structure;
- the vertical structure;
- the horizontal position of the focal company within the end points of the supply chain.

Referring to Figure 1.3, horizontal structure indicates the number of levels throughout the supply chain. Greater is the number of tiers, longer will be the supply chain. It depends on the type of product or service that typifies the specific supply chain. On the other hand, the vertical structure consists of the number of suppliers or customers present in each tier. Larger is the number of organizations at each level, wider will be the vertical structure of the supply chain. Finally, the horizontal position denotes the relative position of the focal company, that can be closer to the initial source of supply or nearer to the end consumer.

Since the network nodes are not all equal and some of them are more critical than other, then also some process links within the supply chain are more significant than others. For this reason, the levels of integration should vary from link to link, and over time (Håkansson and Snehota, 1995). Between the companies that form a supply chain, four different types of business process links exist (Cooper and Lambert, 2000):

- managed business process links;
- monitored business process links;
- not-managed business process links;
- not-member business process links.

Managed process links consist of the set of relations that the focal company identifies as critical to integrate and control. Monitored process links are perceived as less significant for the focal company than the managed ones. The only actions that the focal company does regarding them are auditing and monitoring about how they are integrated and handled among the other actors of the supply chain. Not-managed process links consist of that connections that are not so crucial for the focal company as to allocate resources for management. Finally, non-member process links are the group of relationships that exist between companies of the focal firm's supply chain and other businesses that are not part of the supply chain. This kind of links has to be taken into consideration since the performance of a node of the network, and consequently of the focal company and the entire supply chain, may be influenced also by connections with other supply chains.

1.2.2 Supply Chain Management Processes

Supply Chain Management is composed by 8 principal processes (Lambert, 2008):

- Customer Relationship Management;
- Supplier Relationship Management;
- Customer Service Management;
- Demand Management;
- Order Fulfilment;
- Manufacturing Flow Management;
- Product Development and Commercialization;
- Returns Management.

These processes present strategical and operational sub-processes. The former confers the framework for how the activities will be implemented, integrating a company with the other businesses present within the supply chain. The latter provides the daily activities with all the steps for the implementation of the strategical sub-processes.

Customer Relationship Management consists of the processes that develop and maintain an efficient network and relationship with customers. Important aspect of this process is to identify the target consumer in order to create market segments and provide customized goods and services. For this reason, specific teams oversee the definition of Product and Service Agreements (PSAs), in order to segment customers, meet the need of key accounts and specify levels of performance. The final goal of Customer Relationship Management is to create client retention, to decrease demand variability and to moderate activities that do not add value to the final products.

Supplier Relationship Management aims to generate and preserve a solid relationship with suppliers. As Customer Relationship Management, also this process is fundamental in the management of a supply chain, since the value of a product or service depends on every stage of the network. Generally, a series of traditional relationships is supported by partnerships with a small subset of suppliers built over time. Supplier Relationship Management is about defining and managing these product and service agreement (Lambert, 2008), in order to achieve a win-win relationship for both parties.

Customer Service Management consists of all the activities that has the purpose to control the PSAs developed in Customer Relationship Management processes. Substantially, this process intervenes where there may occur a deviation from the terms defined in the agreement, in order to solve issues before they affect the consumer.

Demand Management consists of all the activities that try to match the customers' needs with the capacities of the supply chain. The process includes forecasting activities which aim to synchronize demand and supply, decrease variability and enhance flexibility. Using customer and point-of-sale data efficiently, it is possible to afford more efficient flows throughout the supply chain.

Order Fulfilment embraces all the activities needed to build an efficient network and meet consumers' expectation while minimizing the total costs. The greater part of this process is performed by the logistics operations, but it also fundamental that it is executed cross-functionally with a proper synchronization of consumers and suppliers. At a strategic point of view, Order Fulfilment applies to themes like service requirements, tax rates, import and export regulations.

Manufacturing Flow Management is introduced in Supply Chain Management in order to reduce undesired inventories, excessive costs and transshipments of goods. It includes all the activities that aim to move materials and goods throughout plants while achieving a certain degree of flexibility in the supply chain at proper costs. The final objective is to reduce cycle times and improve responsiveness to market changes and customers' expectations. To achieve this flexibility level, planning and execution have to be broadened along the supply chain.

Product Development and Commercialization consists of the activities that develop and bring goods to market jointly with consumers and providers. This process supports all the commercialization activities prior the release of the products, and it manages an efficient stream of these products throughout the supply chain once they enter the market. Product Development and Commercialization coordinates with Customer Relationship Management, Supplier Relationship Management and Manufacturing Flow Management processes, in order to identify consumers' needs specifically, select materials and suppliers and develop production technologies.

Returns Management is the collection of activities that aims to manage the reverse flows of goods, identify occasions to reduce undesired returns and detect the possibility of waste reduction. It also embraces some processes of reworking, reconditioning or scrapping if the products simply present defectiveness. This process is fundamental to gain a sustainable competitive advantage.

1.2.3 Supply Chain Management Components

Supply Chain Management Components represent the third factor that characterize the Supply Chain Management framework. They are the elements to integrate and manage a supply chain. They also influence the control of business process links, since the number and level of components included into a business process link determines the level of integration and management of this process link (Ellram and Cooper, 1990; Houlihan, 1985).

For an efficient management of a supply chain, nine different components are distinguished (Cooper and Lambert, 2000):

- Planning and control;
- Work structure;
- Organization structure;
- Product flow facility structure;
- Information flow facility structure;
- Management methods;
- Power and leadership structure;
- Risk and reward structure;
- Culture and attitude.

Planning and control are the activities that aim to develop business links to achieve the supply chain success. Control operations permit the measurement of this success in terms of metrics. The work structure is the method followed by the company to complete its activities and tasks. Organizational structure may indicate the specific company or the entire supply chain. The presence of processes that extent over the firm margins indicates a more integrated supply chain. Product flow facility structure refers to the network structure for providing, manufacturing, and delivering throughout the supply chain. The information flow facility structure represents the type and frequency of information that flow along the supply chain. Management methods consist of the techniques applied in management and the corporate philosophy, like for example top-down or bottom-up structures. Power and leadership structure is an important aspect that influence the configuration of the supply chain and guide its functioning. The same occurs with risk and rewards structure, that determines the engagement of other participant in the supply chain network. Finally, culture and attitude is the component that permits to create a uniform substrate of ideas and attitudes between the various participants, thus allowing the chain to act in a solid and effective way.

1.3 *The importance of Supply Chain Management*

The final purpose of Supply Chain Management is to guarantee the success of a company and to improve the customer satisfaction. Various are the elements that underline the importance of Supply Chain Management (Fawcett et al., 2008):

- Increased inventory turnover;
- Increased revenue;
- Cost reduction;
- Decreased cycle times;
- Improved product availability;
- Amplified market responsiveness;
- Better capital utilization.

The three main benefits are identified in increased inventory turnover, increased revenue and cost reduction. An effective management of the supply chain could result in faster product flows within the network, causing also an increase in cash flows and profitability. A better management of material flows permits the decrease of expensive inventories, thus the reduction of purchasing costs. Furthermore, an efficient supply chain facilitates the diminution of production costs avoiding shortages of materials that would affect negatively the manufacture of products. All these aspects result also in the reduction of total supply chain costs and enhancement of profits, enabling a company to gain competitive advantage in the market.

Other advantages comprise diminution in order cycle times, increased product availability and amplified market responsiveness. An effective management of materials flows and inventory results in reduced cycle time of products. Customers expect the highest precision in terms of product variety, quantity delivered, delivery time and after sale support. A proper management of the entire supply chain is able to improve customers' satisfaction and enhance consumers' fidelity over time. In addition, a strong relationship with suppliers permits to overcome uncertainty in products' demand and to continue to satisfy customers' expectations.

An additional benefit is recognized in capital utilization. A proper control of the network and processes allows the reduction of fixed assets such as manufacturing plants, warehouses and vehicles. An efficient restructuring of the network may translate in significant savings. Cutting superfluous costs, companies within the supply chain may enhance profits.

2. Digitalization and standardization of practices

2.1 Industry 4.0 and supply chain

2.1.1 Supply Chain 4.0

Supply chain has changed during the years, and modern supply chains present a structure and a degree of complexity greater than the older ones. This is caused by events like globalization of markets, intensification of material flows and changes in customers' expectations, mainly caused by the advent of e-commerce. Furthermore, the diffusion during the last decades of more sophisticated information technology systems has affected the supply chain and its management. These new technologies allow to overcome the only possibility to connect humans and machines in a cyber-physical systems (CPSs), enabling also the direct communication between machines. The implementation of this type of network in manufacturing and operations environment is called Industry 4.0 (Tjahjono *et al.*, 2017). The term Industry 4.0 concerns different revolutions in digital technology. Examples include advanced robotics and artificial intelligence, sophisticated sensors, cloud computing, the Internet of Things (IoT), data capture and analytics, digital fabrication (including 3D printing), software-as-a-service and other marketing models, smartphones and other mobile devices, platforms that use algorithms to direct motor vehicles (including navigation tools, ride-sharing apps, delivery and ride services, and autonomous vehicles). Industry 4.0 also consists of embedding all these elements in an interoperable global value chain, shared by many companies from many countries (Geissbauer *et al.*, 2016). Furthermore, Industry 4.0 includes "the use of big data, IoT and Artificial Intelligence (AI) as one" (Tjahjono et al., 2017). The communication between smart machines permits not only the automation of processes, but also the analysis and resolution of various production issues with a lower degree of human involvement.

Another definition of Industry 4.0 given in literature is that "Industry 4.0 is the sum of all disruptive innovations derived and implemented in a value chain to address the trends of digitalization, autonomization, transparency, mobility, modularization, network-collaboration and socializing of products and processes" (Pfohl *et al.*, 2015). The digitalization process is one of the most important aspects that typify the era of Industry 4.0. The internal procedures of the companies, communication channels and other main features of the supply chain are facing a process of through digitalization. Autonomization consists of the process of making something autonomous. The technologies developed permit companies to adopt man-made algorithms to take decision and perform activities independently. Furthermore, the new technologies implemented allow an increased transparency throughout the entire supply chain, making the processes more efficient. Communication channels are becoming even more flexible, consenting interaction and data sharing easier and effective, both between people and among machines in the production processes. The advent of the new technologies enables the modularization of products, which consists of identifying which aspects and parts of a product may be subject to customization and maintain the remaining parts of the product as standard as possible. Modularization results in an augmented flexibility of the production processes and in the possibility to create innovative products while preserving low production costs. In addition, the procedures and activities are determined by the interaction of human beings and machines among specific networks which may extend inside and outside the company's administrative frame. For this reason, the increased interactions in networks aid machines to interconnect with other machines and human beings in a socialized way.

2.1.2 The impacts of Industry 4.0 on the supply chain

It is important to understand which are the direct impacts of Industry 4.0 on the supply chain. Four main areas that may be affected are identified. These areas are:

- Warehouse;
- Transport logistic;
- Procurement;
- Fulfilment functions.

The main areas that are influenced by the diffusion of Industry 4.0 are the order fulfilment and warehouse practices, both in terms of opportunities and threats. Furthermore, the application of some new technologies like virtual and augmented realities, 3D-Printing and simulation represent only a source of chance for the supply chains. On the other hand, big data analytics, cloud technology, cybersecurity, the IoT, miniaturization of electronics, AIDC, RFID, robotics, drones and nanotechnology, M2M and BI represent opportunities and threats for the companies. This double effect is caused by the fact that the different areas of supply chains are interconnected, and a technology may assume a positive or negative meaning depending on the point of view. Despite that, generally it is possible to recognize some specific advantages that generates from the characteristic of the Industry 4.0. The principals are flexibility, quality standards, efficiency and productivity (Tjahjono et al., 2017).

Different point of view suggests that the supply chain may be seen under a vertical and horizonal dimension. The vertical dimension reflects the core operations of the supply chain, that is procurement, production, distribution and sales. On the other hand, the horizontal dimension describes more the organizational perspective of the supply chain in terms of people, structure, technology and people. The variable structure refers to the communication, authority and workflow systems, the variable technology represents assets and the variable people indicates the human domain (Pfohl *et al.*, 2015). Focusing the attention on the horizontal structure, Industry 4.0 has different impacts on the various elements of the vertical dimension. In particular, it has been noted that the supply chain "will undergo an organizational change mainly with respect to the production and distribution processes from a structural perspective" (Pfohl *et al.*, 2015), mainly caused by M2M-communication and the concept of Smart Factory and Smart Logistics. The concept of Smart Factory indicates the application of the new technologies to a manufacturing environment resulting in information transparency and autonomization, while in Smart Logistics such technologies are applied to increase the efficiency in logistics processes like transportation, warehousing and storage. For what concern the technology variable, clearly the implementation of Industry 4.0 will cause a technological change with greater effects in procurement, production and distribution processes. This revolution is principally caused by Smartphone apps, AIDC and RFID technologies and the miniaturization of electronics (Pfohl *et al.*, 2015). Finally, for what concerns the variable people, the greater effects of Industry 4.0 may be seen in sales activity, due to the usage of Smartphone Apps and Smart Data tools (Pfohl *et al.*, 2015).

2.2 *Standardization and automation of processes*

2.2.1 The concept of standardization

In the past years, the changes in the markets, the increased competition and the technological revolution have generated repercussion in the way companies and supply chains make business. In particular, these aspects have made difficult for organizations to compete on price alone. Furthermore, the research of market responsiveness maintaining profitability and competitiveness has led businesses to seek and develop alternative effective and efficient solutions to design, produce and promote a high variety of profitable and worthwhile products (Gudmundsson *et al.*, 2004). In addition, if a company aims to succeed and be competitive within the market, it has to be customer oriented and flexible to the oscillation in the necessity of the markets (MĺÍkva *et al.*, 2016). A possible way to achieve these purposes may be found in standardization. “Standardization is the activity of establishing and recording a limited set of solutions to actual or potential matching problems directed at benefits for the party or parties involved balancing their needs and intending and expecting that these solutions will be repeatedly or continuously used during a certain period by a substantial number of parties for whom they are meant” (Vries, 1999). Consequently, a standard consists of an "approved specification of a limited set of solutions to actual or potential matching problems" (Vries, 1999). In order to enhance their business chances and performance, a lot of companies are investing resources to standardize their business processes. A business process consists of “several sub-processes or activities that are (logically) ordered, having clearly identified inputs and outputs trying to achieve a defined business goal” (Münstermann and Weitzel, 2008). The impacts of process standardization may be found in better processes performance, enhanced readiness, increased ability to react to regulatory changes, improved technical interchangeability and superior customer confidence (Münstermann and Weitzel, 2008).

2.2.2 Principal applications of industrial automation

The advent of Industry 4.0 is also associated with the improvement and optimization of supply chains in terms of automation. This automation affects various supply chain processes and includes software and technology like inventory control, RFID software, Warehouse Management Systems database, Enterprise Resource Planning database and systems that permits warehouse automation and data collection. The integration of these technologies in the supply chain guarantees the accuracy of the activities and operations, also enabling companies to adhere to standard procedures. The importance of automation is particularly recognized in the logistic industry, since it is now integral part of warehouses and distribution centers. “Warehouse-based stockpiling of inventory has been transforming into high-velocity distribution centers, which are increasingly considered strategic to providing competitive advantage. Industry 4.0 can aid the distribution center evolution, enabling adaptable, automated systems that can work with humans” (Taliaferro *et al.*, 2016).

Industrial automation finds application in the following processes:

- Warehouse activities management;
- Intralogistics and automatic handling of loads;
- Planning and management of transports.

Software and machines facilitate the organization of warehouse inventory operations. The optimization of the orders management depends in great measure from the presence in a warehouse of technologies that support the picking activities.

Pick-to-light is one of these technologies (see Figure 2.1). Pick-to-light is a picking technique generally applicable in high-density order picking warehouses with multiple picking locations. Conventional pick-to-light systems store momentarily materials to be picked in flow racks, shelves or workstation, with the products contained in boxes, totes or container, depending on the structure of the system itself. Peculiarity of pick-to-light systems are light devices installed at item locations and used to guide operators in picking operations. When a customer order arrives, the employee scans a barcode on a tote that will contain the customer order. The lights situated on shelves illuminates one at the time indicating the item to be picked and the relative quantity. To confirm the completion of the operation, the picker has to press a specific button or the lighted indicator. When no further lights turn on, picking operation can be considered ended. Picking totes containing customer orders are typically supported on roller conveyors, allowing the employee to move along the shelves.

In high-density order picking warehouses where the presence of human error may be significant, this technology can be implemented to enhance capabilities of employees and improve the accuracy of the picker to 50-99.9% in relation to manual methods (Qinghua *et al.*, 2010). In addition, pick-to-light can be useful to obtain real-time feedback on employees' productivity and progress of picking orders with subsequent improvement solutions. Companies where this system is implemented expect to reach a rate of 450 picks/hour by each operator, value approximately ten times higher than warehouses based on paper-based systems (Murray, 2018).

Finally, pick-to-light can be integrated with ERP or WMS systems already existing within the company. ERP refers to Enterprise Resource Planning. It indicates a typology of software that organizations use to manage different business processes. It ensures a proper and correct data flow between them, providing data integrity and elimination of data duplication. On the other hand, WMS stay for Warehouse Management System and it is a software solution used to manage a business' entire inventory and supply chain fulfilment operations.

Figure 2.1 - Pick-to-Light system

Automation allows greater velocity in incoming and outgoing material flows, thanks to solutions generally implemented in automated warehouses. Examples of the most used automated storage and retrieval solutions are:

- Mini-load stacker cranes;
- Shuttle-based systems.

Mini-load stacker cranes are machines designed for automated storage and retrieval of lightweight loads for various types of cases, containers and boxes through automated movements (see Figure 2.2). They are generally composed by one or two columns, which represent the load-bearing element of all the structure. Column confers the horizontal movement with a specific horizontal speed thanks to a base track with sliding wheels. Connected to the column there is a mobile frame comprising the load lifting device. There are different types of load lifting mechanism like telescopic forks, single bench, grips or drives. This mobile frame is characterized by its own vertical speed. The combination of the movements of these two elements guarantees simultaneous translation along horizontal and vertical direction. Correct positioning within the shelves is made possible by the presence of proximity sensors.

This solution represents one of the possible means of transport suitable to allow high levels of productivity. It suits perfectly in projects when high-paced storage and picking operations are needed. Furthermore, mini-load stacker crane represents a high-density storage solution in relation to a contained occupation surface, also allowing double depth storage. It also reduces errors and enhances data flows providing real-time inventory. Finally, mini-load is a solution which guarantees maximum ergonomics and safety. This is a “product-to-person” solution, where picking operations are conducted out off the shelves, in special locations where the materials are temporarily deposited.

Figure 2.2 - Mini-load stacker crane

Shuttle-based storage and retrieval systems (SBS/RS) consist of a high degree automation solution which is employed in contexts when there is a need of short response time in handling orders, that is when demand for the throughput capacity is high (Lerher, 2016). They found application when the materials handled are lightweight and they are stored in containers, totes, bins or trays within a rack system. Some vehicles called shuttles are employed for the storage and retrieval operations. They move horizontally along the aisles and cross-aisles of the rack system transporting the materials and storing/retrieving them thanks to a load handling device. Generally, there is one shuttle for each tier of the rack. The materials move vertically between the levels thanks to an elevator with a lifting table. In SBS/RS, the elevator often represents the bottleneck. To increment throughput capacity, there is the possibility to install two independent lifting tables. Furthermore, each tier is characterized by buffer positions that work as interface between the elevator and the shuttles. Finally, input/output conveyors connect the elevator with external areas and processes (see Figure 2.3).

SBS/RS can be classified in relation to the degree of freedom of the vehicles. If the vehicles can use the elevator to change tier, they are referred to as tier-to-tier vehicles. On the other hand, if the vehicles cannot change tier using the elevator they are referred to as tier-captive vehicles. As a consequence, in the second case the elevator is only employed in the movement of materials and each level presents a shuttle. Another classification depends on the possibility to change aisle. If the shuttles can change aisle using cross-aisles they are referred to as aisle-to-aisle vehicles. On the other hand, it the vehicles cannot change aisle they are referred to as aisle-captive vehicles. Last classification relies on the levels served by a vehicle. If the shuttles can provide multiple levels, they are referred to as multi-level shuttles. On the other hand, if the vehicles can provide only one level, they are referred to as single-level shuttles.

In comparison with mini-load automated storage and retrieval system, SBS/RS enable higher throughput capacity with capital and maintenance costs relatively greater (Lerher, 2016).

Figure 2.3 - Shuttle-based storage and retrieval system

Different are the automated technologies for the handling of materials that find application in warehouses. The most widely employed are:

- Conveyor systems;
- Automated guided vehicles (AGV).

Conveyor systems are automatic and mechanical handling apparatus used for transportation of loads and materials within a specific area (see Figure 2.4). Generally, conveyor systems allow the transfer of materials from one point to another. They also may move objects among different floors and permit automatic unloading at destination. There are various type of conveyor systems, but generally they consist of a frame that supports rollers, wheels or a belt. Conveyors may be actioned by a motor, by gravity, or manually. They found application when the items to be moved are too heavy for human operators and when there is a necessity to accelerate the operations of transportation. Conveyors systems are distinguished from each other according to some specifications, such as load capacity per unit length, maximum load capacity, speed, throughput, frame configuration and drive location.

The different types that find application in warehouse contexts are:

- Belt conveyors, where materials are transported on a continuous belt made by rubber, plastic, leather, fabric, or metal and extended in an endless loop between two pulleys. Generally, the belt is supported by a metal slider pan, in order to reduce friction. The movement is conferred by motors that use variable or constant speed reduction gears. The conveyor may operate horizontally or inclined.
- Roller conveyors use a set of not powered parallel rollers installed in frames. They convey materials thanks to gravity if inclined or manual action if mounted horizontally. This typology found large application in material handling context, such as picking systems, loading docks, or assembly lines.
- Powered Roller conveyors employ powered rollers installed in frames. The most widespread drive types include belts, chains/sprockets, and motorized rollers. They found application where there is the necessity to automatically convey materials.
- Chute conveyors permit to convey materials to a different level thanks to the effect of gravity. Typical applications of this type of conveyors are scrap handling, or packaging. The maximum efficiency of chute conveyors is reached when the coefficient of dynamic friction is low, allowing the material to move easily without hindrance.
- Ball Transfer conveyors employ a set of unpowered ball casters to convey materials multi-directionally. They are often used in assembly and packaging lines. They found a useful application in connecting different lines, allowing an efficient transfer and sort of products.
- Wheel conveyors are composed by a set of unpowered wheels that allow the handling of materials by gravity or manual power. Generally, they are used for loading and unloading trucks and moving packages or pallets along assembly lines or within a facility.
- Vertical conveyors are used to move materials and products among levels of conveying lines.

Figure 2.4 - Conveyor system

Automated Guided Vehicles (AGVs) are a material handling technology that allows the transportation of objects, travelling in an autonomous way throughout a specific area of a warehouse, distribution center or manufacturing facility (see Figure 2.5). The application of AGVs allows efficient and cyclic movements within the production operations. AGVs are employed for the transfer of raw materials, work-in-process goods and finished products, in order to support the production processes. The last application of AGVs regards storage or retrieval activities.

There are different typology of automated guided vehicle, that differ in relation of the structural characteristics of the vehicle itself and the application fields:

- Automated Guided Carts (AGCs) are the basic type of AGV and they are employed in sorting, storage, and cross-docking operation, with the possibility to handle different type of materials, from small object to loaded pallets.
- Forklift AGVs are the automatic equivalent of the classic fork vehicles used in pallet transportation.
- Towing AGVs employed in the pulling of one or more load carrying vehicles in a train configuration. For their characteristics, they perfectly fit in situations where there is the need to transports heavy loads for long distances.
- Unit load handlers are employed to handle discrete loads, such as single objects or a single carry unit (pallet, tote) containing different items.
- Heavy Burden Carries are used for the heaviest loads in cases like big assembly, casting and coil and plate transport.
- Autonomous Mobile Robots are the more advanced type of AGVs since they incorporate the most sophisticated technology that allow them to dynamically move in the designed area planning the most efficient paths.

Figure 2.5 - Automated Guided Vehicles (Unit load handlers)

The functioning of AGVs is quite complex, since they move and perform operations in an environment populated probably by other vehicles, instruments, machineries and humans. The navigation of AGVs is guaranteed by different type of systems. The most used are:

- Magnetic guide type, where AGVs use magnetic sensors and follow a path constituted by a magnetic tape.
- Wired navigation, where AGVs can move along a path thanks to the signal emitted by wires allocated in the facility floor, which is received by sensors or antennas installed on the vehicle.
- Laser target navigation, where the movement of AGVs is permitted by a laser signal emitted by a transmitter embedded in the vehicle. This signal bounces on a reflective tape mounted on the objects present in the factory floor and it is taken back by a receiver on the AGV. A software is able to interpret the returned signal and to calculate the angle and distance of the object from the vehicle.
- Inertial (gyroscopic) navigation, which employs a computer system and transponders installed in the facility floor to control the movement of AGVs.
- Vision guidance, that permits the movement of the vehicles through cameras, that records the characteristics of the route which are followed by AGVs to navigate. This typology of navigation system does not require the installation of specific infrastructure into the facility floor.
- Geoguidance, where AGVs are able to recognize the objects in the area, determine their actual position and navigate through the building. Geoguidance does not need the installation of infrastructures.
- LiDAR (Light Detection and Ranging), which consists of an advanced technology. A laser transmitter emits pulses that are used to create a 360-degree map. This allows the vehicles to calculate the distance from the objects and move throughout the facility. Also in this case, there is no need to install supplementary infrastructure.

The possibility to change direction along the path is guaranteed by the AGV steering system. It is controlled by different solutions:

- Differential speed control, which is the most applied control. It functions with two independent drive wheels. The movement of the vehicle depends on the relative velocity of the two wheels. If the two wheels rotate at the same speed, the vehicle moves in a straight line forward or backward. On the other hand, if the two wheels rotate at different speed, the vehicle is able to turn. This control system is generally employed in cases where the AGV moves in tight spaces or works near machines.
- Steered wheel control, where the turning movement is conferred by the turning wheel. This system is more accurate than differential speed control and guarantee a smoother curving. This control method is often used for towing AGVs.
- Combination steering, which consists of a mix of the two control systems described above. AGV have two independent motors or drive wheels positioned on the diagonal corners of the AGV and two swiveling castors on the other two corners. This configuration allows movements in any direction.

In addition, automated guided vehicles present traffic control systems. Three different typologies exist:

- Zone control, which is largely employed. A wireless transmitter sends signals in specific areas. The AGV picks up the signal and sends it back to the transmitter. A free signal is transmitted to the AGV if there is not another AGV in the area, while a stop signal is transmitted if another AGV is moving in the area. In the second case, the AGV waits until the area becomes free.
- Collision avoidance, which consists of sensors that avoid the collision of the vehicle transmitting a signal in order to identify the presence of an object. The sensors used may be sonic, optical or contact.
- Combination of systems, that consists of a mix of the two technologies described above. This typology confers a more performant collision prevention. Generally, zone control represents the main traffic control system, while collision avoidance supports in the event of a failure of the primary system.

Many are the benefits that generate from the introduction of AGVs in warehousing and manufacturing contexts. They increase the efficiency and productivity, since they operate autonomously and may be employed for repetitive tasks. In addition, they reduce the physical labor in load transportation to minimum levels, they enhance accuracy, they decrease the errors deriving to human actions and they are a safe solution for warehouses, distribution centers and production facilities. In relation to the high variability in human labor costs, they are affected by less cost fluctuations since they are typically acquired on a per unit or per rental period cost basis. AGVs also guarantee a high level of flexibility. In fact, they are able to change routes easily and they represent a scalable solution, simply adaptable to variation in market demand. Furthermore, in comparison to other automatic solutions or traditional systems, AGVs are smaller and they need less space.

The last processes in which industrial automation finds application is transport planning and management. Thanks to the implementation of specific software, it is possible to manage the fleet and to better coordinate the distribution phase of products. These systems allow routes to be planned according to different logistic parameters, such as delivery times or shipping techniques used. The increased data quality and availability are a basic aspect of Industry 4.0, which has repercussion on transportation planning. For example, data quality and transparency permit to create more accurate shipping and demand forecasts. In addition, a high degree of flexibility is added, since shipping and routing options may be change on a real-time basis in relation to particular data, such as traffic, wheatear, freight capacity and warehouse utilization.

2.3 The Digital Twin

With the diffusion of the idea of Industry 4.0, another concept has begun to be used in the recent years: the Digital Twin. “The Digital Twin (DT) is one of the main concepts associated to the Industry 4.0 wave. […] The Digital Twin is meant as the virtual and computerized counterpart of a physical system that can be used to simulate it for various purposes, exploiting a real-time synchronization of the sensed data coming from the field; such synchronization is possible thanks to the enabling technologies of Industry 4.0”. (Negri *et al.*, 2017).

From this definition, it is possible to understand which are the principal characteristics of a Digital Twin. A Digital Twin is formed by three elements, namely the physical entities in the physical system, the virtual entities within the virtual system, and the data that connect the two worlds (Tao and Zhang, 2017). “The data from physical world are transmitted to the virtual models through the sensors to complete the simulation, validation and dynamic adjustment. And the simulation data are fed back to the physical world to respond to the changes, improve the operation, and increase the value” (Qi and Tao, 2018). Industry 4.0 brings the growth and advancement in technologies, which permit interconnection and interaction between advanced components. This aspect allows the closed-loop communication between the real world and the virtual world and the optimization of processes. “Simulations will leverage real-time data to mirror the physical world in a virtual model, which can include machines, products, and humans. This allows operators to test and optimize the machine settings for the next product in line in the virtual world before the physical changeover, thereby driving down machine setup times and increasing quality” (Lorenz *et al*., 2015).

Furthermore, from the definition given above, the Digital Twin is used for simulating a real system for various purposes. Nevertheless, Digital Twin overcome the concept of simulation (see chapter 3 - Simulation modelling). Indeed, “in both cases, the simulation is happening on a virtual model, but the model becomes a digital twin once the product is produced. When a digital twin is powered by an Industrial Internet of Things (IoT) platform, it can receive real-world data quickly and process it, enabling the designer to virtually see how the real product is operating” (Maloney, 2019).

The horizon of this research stops to the simulation aspect of a Digital Twin, without takin into consideration the real-time connections between the actual and the virtual system.

3. Simulation modelling

3.1 Introduction to simulation modelling

The concept of simulation modelling grounds on the ideas of models and simulations. “A model is a representation of the construction and working of some system of interest” (Anu, 1997). Moreover, it “is a simplified representation of a system at some particular point in time or space intended to promote understanding of the real system” (Bellinger, 2004). Thus, a model can be considered as a simplification of the system it tries to replicate, even though it has to incorporate all the principal characteristics of it.

Models can be classified in relation to different aspects:

- Physical / virtual
  The replication of the system into analysis may be done with the construction of a smaller physical model. An example of physical model can be the aerodynamic tests made into wind tunnel on small objects with the same characteristic of the real one. On the other hand, a virtual model does not employ physical items, while utilizes a specific software able to replicate features and performance of an object in a computer-generated environment. The categorization of models into physical or virtual principally relies on the system to model.
- Deterministic / stochastic
  This classification depends on the value that variables assume within the model. In particular, models can be divided into deterministic, when input and output variables are fixed values, or stochastic, when at least one of the input or output variable assume a probabilistic value (Anu, 1997). A deterministic model is typified by rules and behaviors that remain constant over time, while a stochastic model exhibits random performance under identical conditions. Elements of randomness characterize systems that are more realistic.
- Static / dynamic
  Models can be classified as static, when time is not considered, or dynamic, when time-varying interactions among variables are taken into account (Anu, 1997). In a static model, all the calculations and actions are performed and then later the result is given as output of the model itself. On the other hand, adding the time component in dynamic models, the complexity of performance of the model rises considerably. The diffusion of dynamic models derives from the fact that an analytical solution, i.e. the one obtained from a static model, may be difficult to find or does not always exist (Borshchev and Filippov, 2004). Dynamic models are also called simulation models.
- Discrete / continuous
  With regards to dynamic simulations, it is possible to classify models into discrete or continuous in relation to time influence of state changes. Into discrete models, state changes occur at specific intervals of moments and what happens into intermediate states is ignored. On the other hand, continuous models simulate system where changes arise perpetually.

A simulation is the representation of the behavior or characteristics of one system through the use of another system, especially a computer program designed for the purpose (Dictionary.com, 2020). Simulation modelling is a discipline that tries to explain how something works by building a replica of it (Mahdavi, 2019), i.e. a model of the analyzed system. The construction of this reproduction consists of a set of rules which describe the changes of the system over time. In fact, simulation is the process of model "execution" that takes the model through (discrete or continuous) state changes over time (Borshchev and Filippov, 2004).

3.2 How simulation modelling works

3.2.1 Simulation modelling framework

Simulation modelling works across two universes, the real world and the modeled world. Real world is the environment where it is possible to find the analyzed system, with all the relative problems to be solved. On the other hand, the modeled world is a risk-free world, through which it is possible to find the solution to the problems detected into the real world. Simulation becomes useful to evaluate an optimal design solution of the system prior to implementation. This can help to reduce chances of over/under utilization of resources, to evaluate the presence of possible bottlenecks within processes and to maximize the overall performance of the system. Finally, simulation modelling plays a fundamental role when an existing system presents some problems to be solved but the execution of experiments directly on it turns out to be too expensive or practically impossible. In these cases, a model of the analyzed system can be built in order to find an optimal design solution at the model level executing experiments on it. Consequently, the solution founded may be applied to the system at real world level. The framework of simulation modelling is depicted into Figure 3.1.

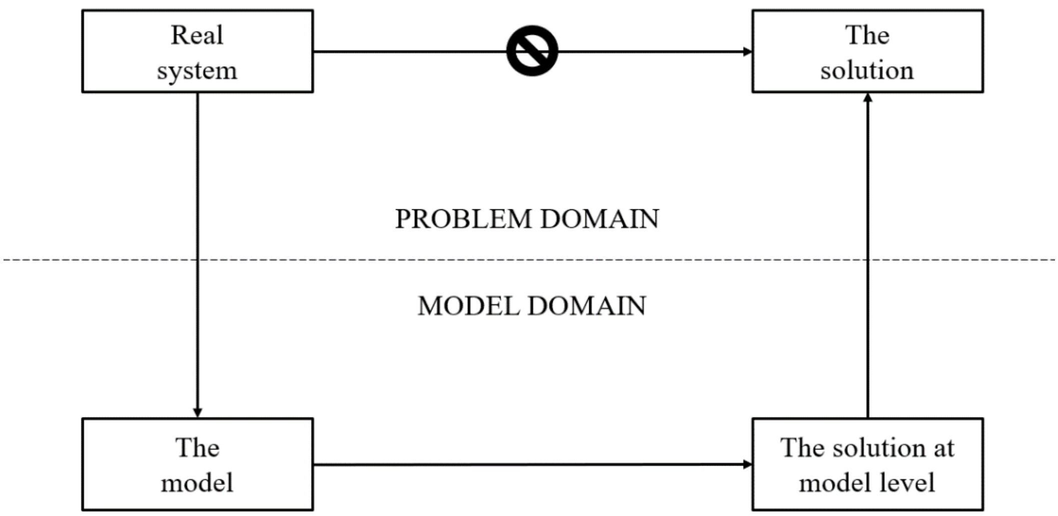


Figure 3.1 - Simulation modelling framework (adapted from Borshchev and Filippov, 2004)

3.2.2 Conceptual modelling

Simulation modelling framework (Figure 3.1) shows a direct connection between the real system into study and the model of it. Nevertheless, the passage from problem domain (real world) to model domain (modeled world) is not so direct, but it includes intermediate steps. These phases consist of what is called conceptual modelling. Conceptual modelling is applied at early stage during project development (Daum, 2003 and Robinson 2011). Depending on the context of simulation modelling, the business domain can be a new system or an existing one with all the relative problems at the basis of the study.

Firstly, it is important to identify which are these problems within the system. This phase is fundamental to understand where the criticalities reside, in order to better focus on them. Indeed, through knowledge acquisition and formulation of assumptions regarding uncertain or unknown aspects, it is possible to move to the system description, which is "a description of the problem situation and the system in which the problem situation resides" (Robinson, 2011). Based on the system description and thanks to the abstraction process, a conceptual model can be built, capturing the business domain with more formal means (Daum, 2003).

The process of abstraction of a real system is the basis for conceptual modelling (Robinson, 2011). It allows the connection between the problem domain and model domain. An abstraction is a general idea rather than one relating to a particular object, person, or situation (collinsdictionary.com). The process of abstraction consists of hiding details or characteristics that can be considered unnecessary and superfluous. It allows "to implement more complex logic on top of the provided abstraction without understanding or even thinking about all the hidden complexity" (Janssen, 2017). Since conceptual modelling involves the introduction of assumptions and the neglection of some detail, it is important to understand the right level of abstraction, which "determines the amount of information that is contained in the model. The quantity of information in a model decreases with the levels of abstraction" (Benjamin *et al.*, 1998). This means that lower is the level of abstraction, higher is the quantity of information contained into the model. Nevertheless, all models contain simplifications at different level of the real world, so "all simulation modelling involves conceptual modelling" (Robinson, 2011). Consequently, establishing the proper level of abstraction is an important aspect of simulation modelling, it is one of the first steps in building a model and it depends on modelling objectives.

Figure 3.2 schematizes the process of conceptual modelling within the wider context of simulation modelling. Conceptual model is the first element of simulation modelling that belongs to model domain. Various are the definition of what a conceptual model is. It "is a solution-independent description of a real-world problem domain, from which a platform-independent simulation design model can be derived for a given set of research questions" (Robinson *et al.*, 2015). Additionally, it is "a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model" (Robinson, 2008). Many information is contained within these two definitions. They both state that the conceptual model is still separate from the computer one, thus it is independent from the solution that can be obtained through a simulation software. Indeed, it only represents the basis for further develop a computer model in a specific computer code. A conceptual model is the description of a computer simulation model "that will be, is or has been developed", thus it is not a static entity, but it changes continually during the building process of the study. From this aspect it is possible to understand the iterative nature of conceptual modelling, thus is not limited to the initial phase of a simulation project but needs to be applied through all life-cycle phases (Robinson *et al.*, 2015). Furthermore, conceptual model includes all the elements that characterize the model itself. It contains the objectives or purposes behind the creation of it, which it is fundamental to identify correctly in order to create a proper simplification of the system into analysis. The conceptual model describes the input and output of the model. Inputs are the elements that are modified throughout simulation modelling in order to attain the objectives identified. On the other hand, outputs are the simulation results, that is statistics which apprise us if modelling purposes have been reached and otherwise the reasons behind the unexpected results. Conceptual model includes also a description of the content of the model regarding the environment modelled and at which degree of detail. Moreover, it illustrates which are the assumptions and simplifications made during abstraction process.

Applying the design process to conceptual model, it is possible to exit from conceptual modelling domain and the model design can be achieved. It consists of the structured creation of objects with the aim of implementing specific features, that is the constructs for the computer model (data, components, model execution, etc.) (Fishwick 1995).

When the conceptual model and the model design are translated into a specific computer code, the computer model is generated. It basically consists of "a software specific representation of the conceptual model" (Robinson, 2011). Unlike this latter artefact, the system description, conceptual model and model design may not be fully explicated and may lie into modelers mind, although it is preferable to document properly each of them (Robinson, 2011).

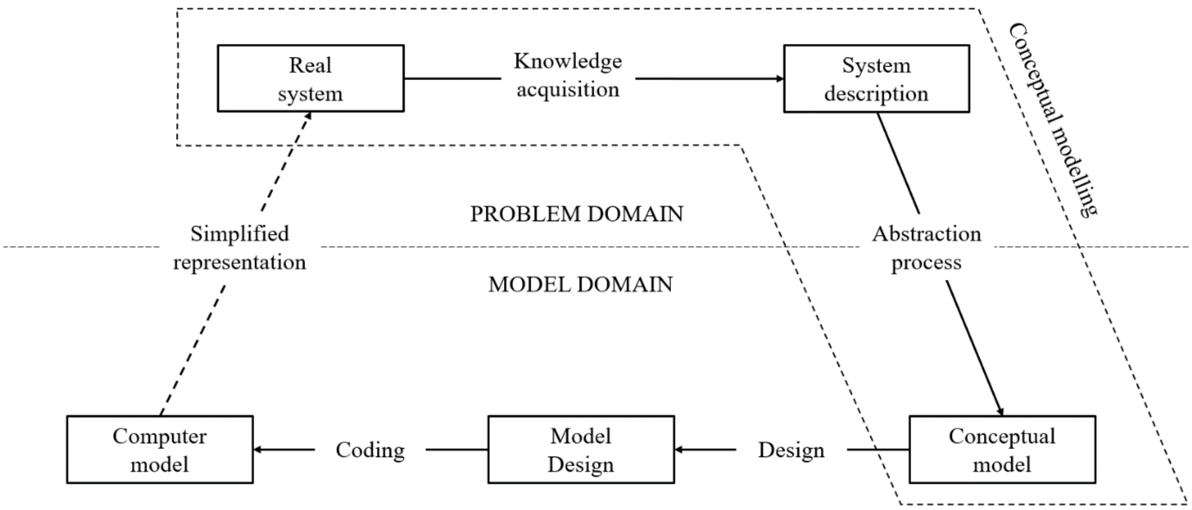


Figure 3.2 - Conceptual modelling within simulation modelling context (adapted from Robinson, 2011)

3.2.3 Model verification and validation

Creating a working virtual model that simplifies the real system into analysis is not the last stage of the modelling process. Once completed, models have to be tested in term of representativeness through the verification and validation processes.

The term verification is used to describe the process that attempts to establish that a computer simulation model is consistent with the underlying conceptual model upon which it is based (Murray-Smith, 2015). Another similar definition of verification states that it "is the process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model" (Thacker *et al.*, 2004). Basically, it consists of comparing the model performance with the model specifications, finding out if exists consistency among them.

On the other hand, "validation is the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model" (Thacker *et al.*, 2004). In validation, consistency is searched among the model performance and the real system simplified through the model. The concept of validation is strictly related to developing a simulation model that possesses an appropriate accuracy consistent with the problem into study (Robinson, 2011 and Sargent, 2011). Affirming the validity of a model as defined above involves the identification of an acceptable range of correctness for each output variable of the model, that is generally specified as the maximum acceptable difference between the model variable and the corresponding system variable to consider the model still valid. This level of correctness should be specified prior the development process or at beginning stages of it (Sargent, 2011).

Figure 3.3 depicts a simplified version of the modelling process and the roles of verification and validation within it.

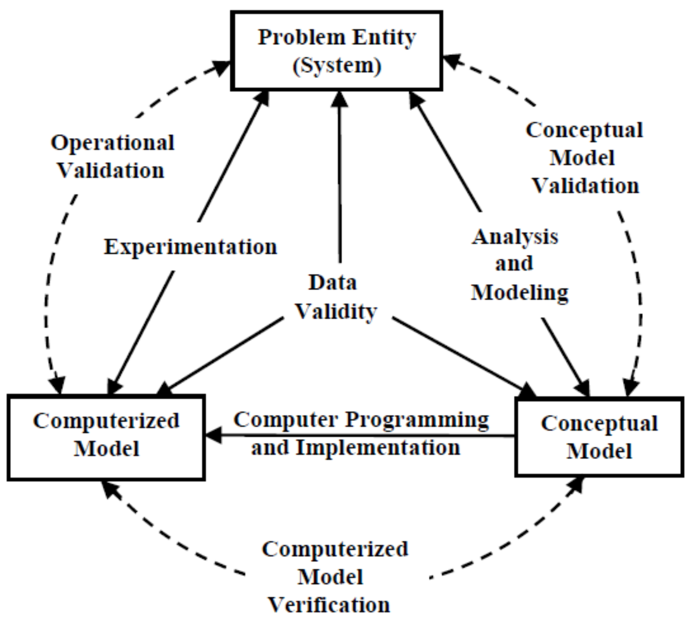


Figure 3.3 - Simplified version of the modelling process (Sargent, 2011)

Modelling process involves three elements: problem entity, conceptual model and computerized model, as defined in sub-sections 3.2.1 and 3.2.2. The simplified version of the modelling process in Figure 3.3 adds the relations among these entities. Conceptual model validation consists of the determination of correctness of the theories and assumptions at the basis of the conceptual model. Computerized model verification aims to guarantee coherence between the conceptual model and the virtual model, that is accuracy in programming and implementation. Operational validation consists of verifying the adequate accuracy of the model's behavior in relation to the application system and the intended purpose of the model itself. All the artefacts of modelling process depend on data validity, defined as the correctness of data necessary for all the stages of development process.

Interesting point of view about model validation and verification is given by Robinson (2011). He finds a relationship between model accuracy and its complexity in terms of scope and level of detail. The attention is placed to conceptual models, but the relation may be extended to simulation model in a more general way. According to literature, the fundamental requirements of a model are identified in validity, credibility, feasibility, and usability. Validity is perceived as described in this sub-section. Model credibility is concerned with developing users' confidence in using the model and believing in it, while feasibility regards the possibility to build the model within the project constraints in terms of due date and time. Finally, usability means that the model has to be flexible, easy to use, visual and quick. Generally, these are the requirements in order to meet the purposes of the study building the simplest model possible. Models require simplicity and simpler models are favorite compared to more complex ones because they can be built faster, they are more flexible, they necessitate less quantity of data, they can be executed faster, and the results are easier to understand. As said before, model structure and characteristics are strictly correlated to simulation objectives. This means that more complex models should not be avoided, but they have to be developed only if needed to achieve modelling purposes. Figure 3.4 shows the relation present between model complexity and its accuracy.

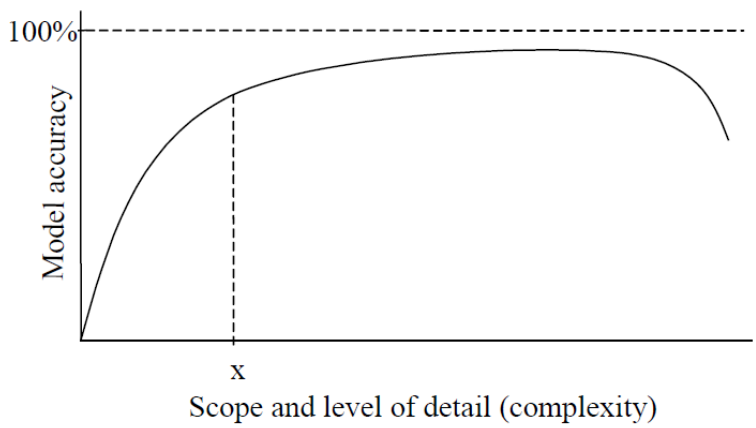


Figure 3.4 - Simulation model complexity and accuracy (Robinson, 2011)

It is possible to notice that with increasing level of complexity of the model also increases the level of accuracy of the model, but never reaching the value of 100% of precision. Nevertheless, the growth rate tends to decline as the complexity of the model increases. In addition, at a specific complexity value, accuracy will stop growing, but instead will begin to decrease. The effect of this phenomenon is attributed to the lack of sufficient knowledge of the system to support the complexity integrated in the model, which lead to the introduction of incorrect assumptions. The issue is finding the proper trade-off between model complexity and accuracy, indicated in Figure 3.4 as x. Moving along positive verse of complexity axis, it is possible to marginally increase the precision of model in relation to the great effort to enhance the level of detail of simulation. Nevertheless, it may be necessary to include more complexity due to study purposes. It may difficult if not impossible to find the best model among an infinite set of possible solutions (Robinson, 2011).

3.3 The principal simulation modelling paradigms

A simulation modelling paradigm is defined as the approach selected to create a simulation model, that is it stipulates the hypothesis, ideas and rules applied in the construction of the virtual replicas. Different are the typologies of paradigms existing in simulation modelling. They differ from each other in relation to the level of abstraction, that is the level of detail contained in simulation. The principal approaches widespread in simulation modelling are three:

- System Dynamics;
- Discrete Event Simulation;
- Agent Based Simulation.

It also exists another paradigm called Dynamic Systems, which it is basically used to model physical system and for this reason it does not apply entirely to the concepts of simulation modelling. In general, Dynamic Systems and System Dynamics manage continuous processes, while Discrete Event Simulation and Agent Based Simulation cope with discrete events.

Regarding continuous simulations, Dynamic System presents the lowest level of abstraction, while System Dynamics is located at the highest position in terms of abstraction. With regards to discrete simulations, Discrete Event Simulation applies in low to middle abstraction cases, while Agent Based Modelling can be used across all the abstraction levels thanks to its characteristics.

These aspects of the different simulation modelling approaches are schematized in Figure 3.5. Dynamic system is not included in the diagram, since it is outside the context of simulation modelling.

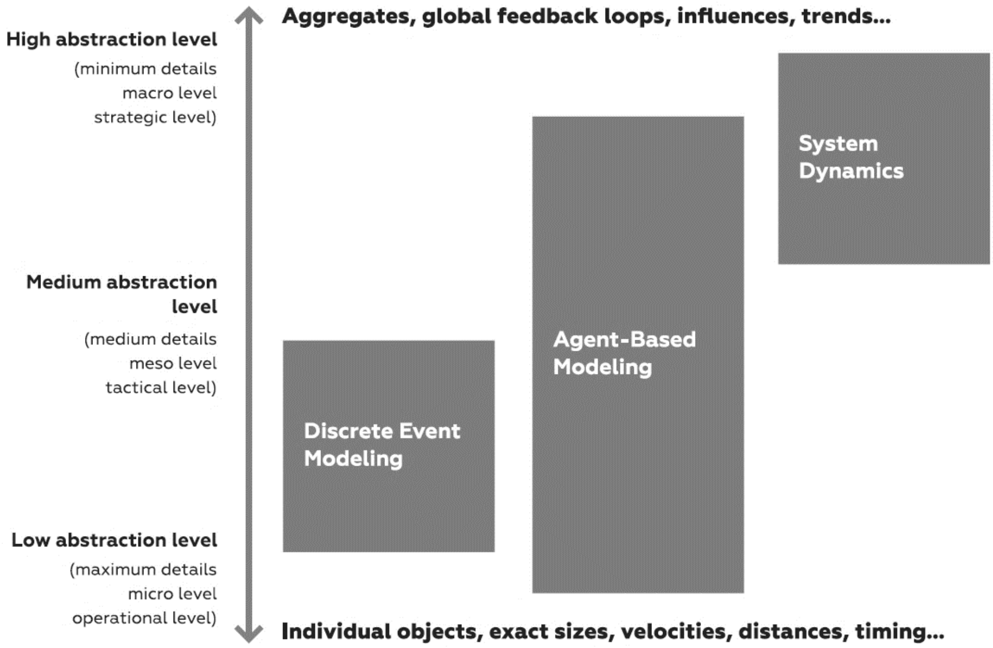


Figure 3.5 - Paradigms in simulation modelling on abstraction level scale (adapted from Borshchev and Filippov, 2004)

3.3.1 System Dynamics

System Dynamics is the highest level of abstraction paradigm in simulation modelling and it offers a method to develop and analyze models that aim to exemplify the dynamic performance of systems over time. It finds its roots in work done by the professor Jay W. Forrester from the Massachusetts Institute of Technology. In 1961 he wrote a seminal book called Industrial Dynamics, which laid the foundations for this paradigm. System Dynamics is "the study of information-feedback characteristics of industrial activity to show how organizational structure, amplification (in policies), and time delays (in decisions and actions) interact to influence the success of the enterprise" (Forrester, 1961). The System Dynamic Society defines System Dynamics as a computer-aided approach to policy analysis and design.

From the beginning, the fields of application were limited to corporate, industrial, organizational and managerial problems. During the years, the paradigm was also introduced in other contexts. In fact, nowadays it is employed in the resolution of dynamic problems occurring in complex social, managerial, economic, or ecological systems, i.e. dynamic system distinguished by interdependence, mutual interaction, information feedback, and circular causality. Thanks to these characteristics, it is mostly used in strategic modelling.

The fundamental element in system dynamics is the stock and flow diagram. A stock is the virtual representation of something. Some example of stock might be money, material, people, products, conditions or knowledge. Since stocks represents group of entities, the elements in the same stock do not possess distinctiveness and are impossible to differentiate. On the other hand, a flow represents a process existing between stocks. In addition, in System Dynamics it is also modelled the information that defines the values of the flows. This information is included in the model thanks to auxiliary variables that generate a causal relationship or through feedback loops, which consist of using the outcome of past performance of the system to impact the future execution. Therefore, in System Dynamics it is necessary to consider existing global structural dependencies and specify reliable data in quantitative and qualitative terms.

From a mathematical point of view, a System Dynamic model is a set of differential equations, which are utilized in order to determine the values of the modeled elements over time.

In Figure 3.6 is possible to observe a simple example of a System Dynamics model.

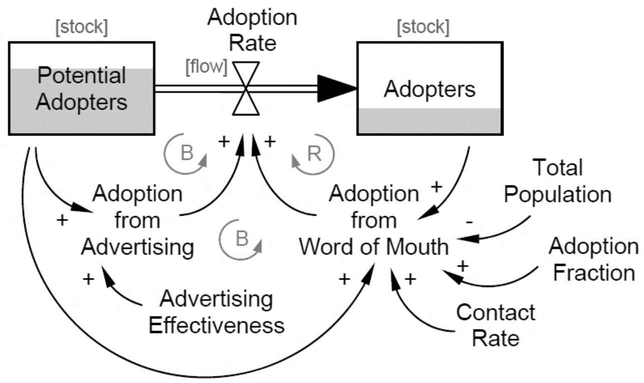


Figure 3.6 - System Dynamics model (Borshchev and Filippov, 2004)

3.3.2 Discrete Event Simulation

Discrete Event Simulation consists of a set of techniques, which applied to the analysis of a discrete-event system generates a model able to exemplify its performance. A discrete event system is a system in which at least one phenomena of interest change value or state at discrete points in time, instead of continuously with time (Fishman, 2001). In contrast to continuous systems, in Discrete Event Simulation it is introduced the assumption that nothing with enough relevancy happens between two consecutive events, i.e. the system into analysis does not change its state between the events. The change in the state of the system coincides with event occurrence. Peculiariarity of this type of simulation is the presence of system state variables, that consist of a set of information which describes the behavior of the system at a specific point in time, that is they define the system state.

The father of Discrete Event Simulation is identified in IBM engineer Geoffrey Gordon (Borshchev and Filippov, 2004), who introduced in the 1960s the first version of General-Purpose Simulation System (GPSS). It is considered the first implementation at software level of the concepts of discrete events modelling. GPSS is a process-oriented simulation language used for modelling discrete systems. Distinctive feature is the presence of a set of standard blocks employed in the construction of models. In GPSS, entities are called transactions and consist of objects representing people, components, products, documents, tasks or messages. Standard blocks permit the definition and control of transactions' behavior. The elements of the model are translated into a GPSS program through the choice of specific blocks. These blocks are interconnected to generate a block diagram that aims to determine the logic structure of the system. By interpreting and executing the chart, the simulation is generated. This phase is particularly slow and for this reason GPSS is applicable only to small-medium dimension problems.

Thanks to its characteristics, Discrete Event Simulation paradigm applies better to processes at medium and medium-low level of abstraction. Therefore, specific physical details of the elements populating the system are omitted. Fields of application of Discrete Event Simulation are inventory systems, manufacturing plants, distribution systems, transportation networks.

In simulation modelling, the performance of these systems is measured in terms of delays, buffer occupancy, throughput and resource utilization (Fishman, 2001). Delay consists of the time spent by an element of the system waiting for resources. Buffer occupancy indicates the number of jobs, items or individual waiting to be processed by resources. Throughput represents the number of completed jobs per time unit, while the time percentage of the busy time in relation to total time defines resource utilization.

Figure 3.7 depicts an example of Discrete Event Simulation.

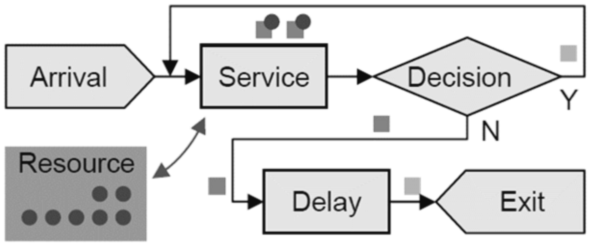


Figure 3.7 - Discrete Event Simulation (Borshchev and Filippov, 2004)

3.3.3 Agent Based Simulation

From revision of literature, it is difficult to establish a unique definition of agent, since this concept apply to different areas of interest. Agents can be seen as active autonomous entities that are situated in an environment where they have possibly restricted perception and local manipulation capabilities. The environmental model sets the frame for the agents (Klügl, 2016). Additionally, an agent can be considered as an autonomous decision-making entity that individually assesses its situation and makes decisions on the basis of a set of rules. The behavior of the agent depends on the system it populates (Bonabeau, 2002).

Despite the absence of a common definition of agent, agent-based models have the common feature to be decentralized (Borshchev and Filippov, 2004). This means that in agent based models the performance of the system replicated is generated by the actions and interactions of a set of agents (Klügl and Bazzan, 2012), instead of merely depend on the description of its global behavior it as it happens in other simulation paradigms. Therefore, the behavior of the system emerges as the result of a large number of individuals living in the same environment, each following its own rules, interacting with each other and communicating with the environment. The dimension of the group of agents clearly depends on the system modelled. As a consequence of the aspects described above, this simulation paradigm is also called bottom-up modelling or microscopic modelling. Thus, Agent Based Simulation is suitable for a wide range of abstraction levels. In fact, this methodology is used to model physical objects, population, road traffic, markets, and supply chains or more in general where there is a need to a major focus on individual objects.

Agent Based Simulation presents advantages and disadvantages (Klügl and Bazzan, 2012). It has the powerful characteristic of allowing observation and study of the dynamics of the model at two level of detail, that is the local agent level and a more macroscopic level determined by the behavior of the agents' population. This feature allows complex design of the agents, or rather there is no limit to complexity of their inner configuration. In contrast to more macroscopic simulation paradigms like System Dynamics, this leads to overcoming the assumptions of homogeneity of the entities living in the model and the introduction of individualities and specific behaviors. Consequently, one of the most important peculiarities of Agent Based Simulation is flexibility. On the other hand, the high degree of versatility might become a critical aspect, mostly in terms of determination of the right level of abstraction and the size of the model. In addition, the development process seems to consist of evaluating different design solutions instead of following a conventional simulation process. Because of that, the proper setting of model parameters might be crucial and minimum changes may lead to a completely different result. Another potential issue with Agent Based Simulation is reproducibility of results and validation process, the former generating from incomplete information about the agent-based model, while the latter caused by a lack of empirical data available in the majority of cases.

Figure 3.8 represents the generic architecture of an agent-based model.

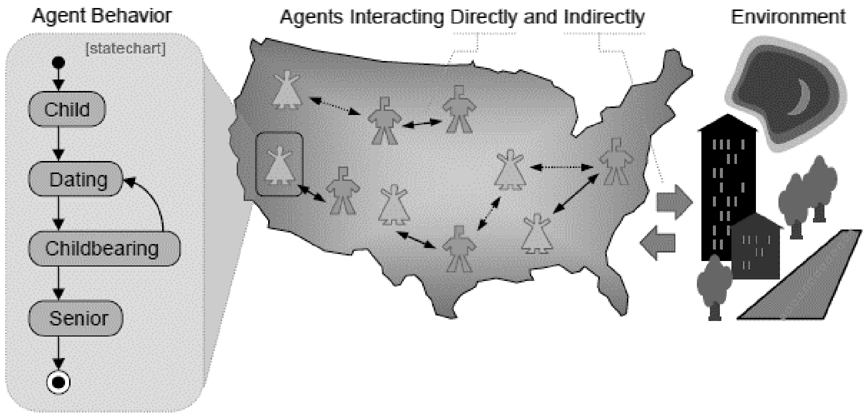


Figure 3.8 - Agent Based Simulation (Borshchev and Filippov, 2004)

3.3.4 Multi-method simulation modelling

Even though this classification of the three main simulation modelling paradigms, nowadays it is possible to witness an increasing use of combined methodologies to cope with complex problems and processes with very different features. In order to overcome the limitations of each paradigm described above, it is possible to integrate the simulation techniques in different part of the model. This approach is called multi-method simulation modelling. The basic principle behind it is to use the advantages of every paradigm, build synergies between them and generate a more efficient and correct model specific for the case. Generally, the combination of methodologies applies in cases when the system into analysis is sufficiently complex to include some aspects and peculiarities of each basis paradigms.

4. The physical system: automated warehouse

4.1 Project description and methodological steps

The project proposed by Polytechnic of Turin involves the creation of a simulation model of an automatic warehouse planned to be built in the future for direct research purposes on a real system. The objective of the study consists of the analysis of the performance of the warehouse at a virtual level, in order to find where reside the criticalities and propose an optimal design solution, overcoming all the complication of a direct hardware implementation. The warehouse will be used for the storage of a limited number of small components, which can be assembled in small products destined to shipping.

Following the concepts and fundamentals of simulation modelling, the research is composed of the following steps:

- Analysis of the case study
- Construction of a conceptual model
- Creation of a virtual model
- Simulation of different experiments varying specific parameters
- Analysis of the results of the simulations

In this chapter, the first step of the research is illustrated, the conceptual model is explained in chapter 5 - The conceptual model, the virtual model and the different experiments are described in chapter 6 - The virtual model, while the analysis of the results is contained in chapter 7 - Experiments and results.

4.2 The warehouse

According to the theories of simulation modelling explained in chapter 3 - Simulation modelling, the first step to follow in order to build a functioning and valid model is the description of the real system. In this case, the system does not exist physically, but there are some project specifications of the warehouse.

The warehouse will be installed in a laboratory that has a rectangular shape with dimensions that amount to 14.53 x 4.74 meters. In addition, the area includes a corridor around 2 m wide positioned along the top long side. The floor map of the area dedicated to the construction of the warehouse is depicted in Figure 4.1 (the corridor it is not included in the map).

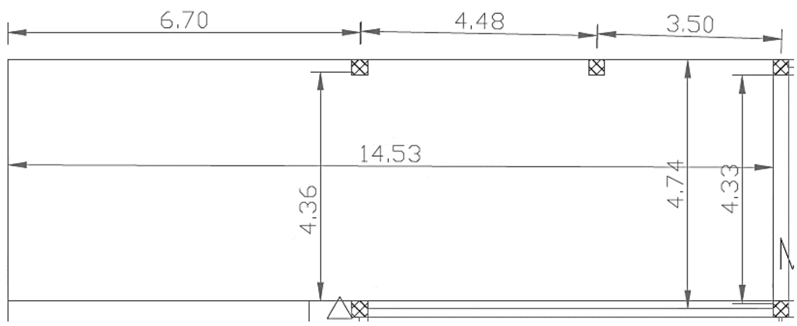


Figure 4.1 - Floor map of the laboratory

The bottom right area is employed for the installation of an automated warehouse, while the left area and the corridor are dedicated to other operations.

The automated warehouse:

- allows the storage of plastic totes and carton boxes with the same dimensions and sufficient firmness;
- has a mini-load stacker crane designated to totes movement;
- includes a manual picking station for the preparation of orders destined for production and/or shipment. The station must have a minimum of stock of empty totes to be filled with production or shipment orders for subsequent movement to and from workstations;
- comprises a workstation for the preparation of kits destined for assembly. The station includes an assisted pick-to-light system;
- has an input conveyor and an output conveyor dedicated to handling by AGVs.

More specifically, the different areas of the automated warehouse have been subdivided as depicted in Figure 4.2:

1. Rack warehouse for totes;
2. Conveyor exit from warehouse to AGV or kitting area;
3. Gravity exit from warehouse to kitting area;
4. Gravity exit from warehouse to picking area;
5. Totes pick-up point by AGV;
6. Kitting workstation;
7. Picking workstation;
8. Picking area totes exchange point (full/empty) from/to other areas;
9. Totes input point by AGV and operator;
10. Mini-load stacker crane that works as interface between rack warehouse and input/output points.

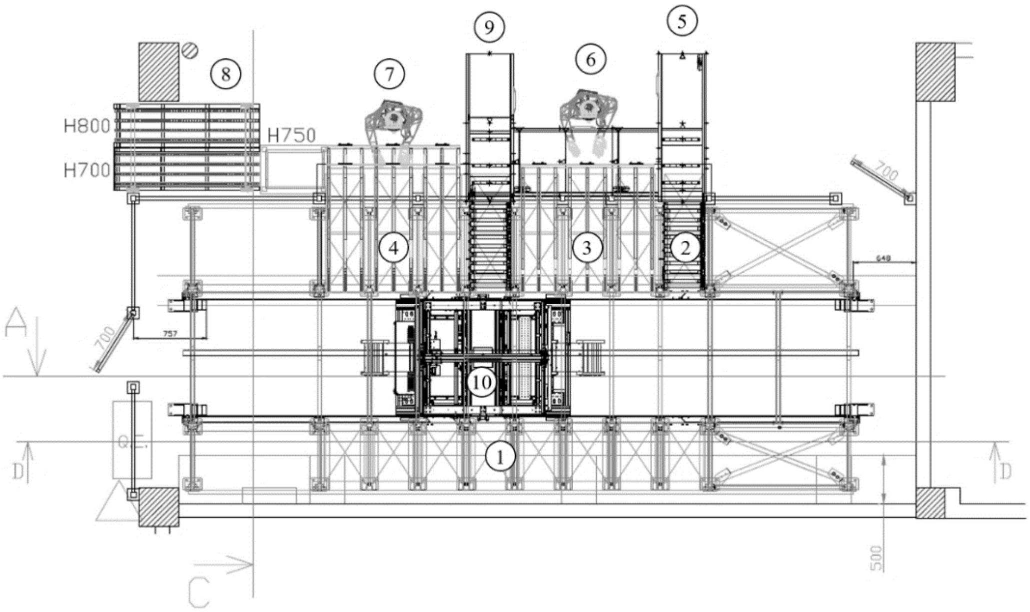


Figure 4.2 – Floorplan view of the automatic warehouse

In Figure 4.3 is possible to see the D-D section (see Figure 4.2) of the automated warehouse.

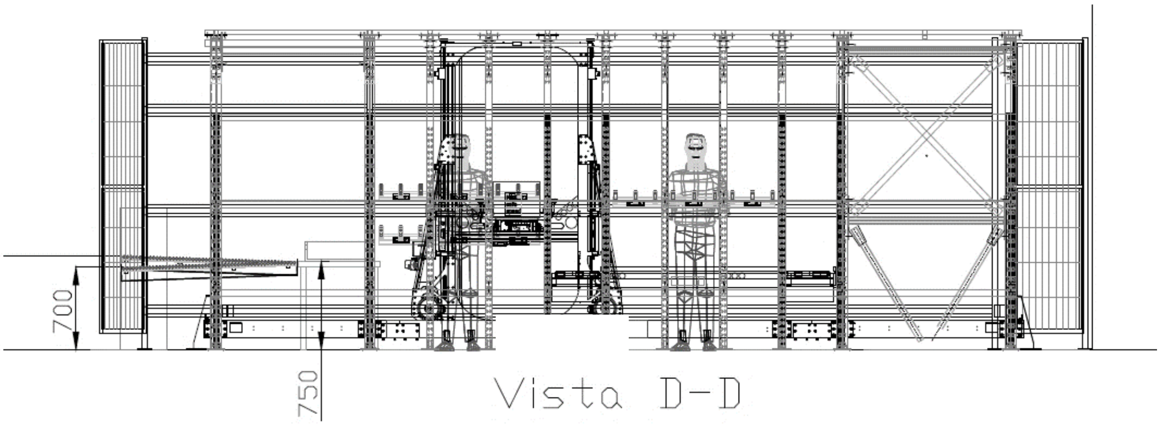


Figure 4.3 - Frontal view of the automatic warehouse

The following are the technical characteristics of the automated warehouse:

- The mini-load stacker crane works on fixed floor rails and it is equipped with a platform for access to the different levels of storage and a load unit gripping tool. The maximum load supported amounts to 50 kg;
- The number of storage cells is equal to 40, subdivided in 1 row, 5 levels and 8 single depth positions;
- The plastic totes and the carton boxes measure 400 X 600 X 175 mm;
- The plastic totes are equipped with metallic supports necessary to insert barcodes for the tracking. Furthermore, the bottom side has to be flat to allow the movement through roller conveyors;
- The warehouse is provided with two entrances of 700 mm to allow the passage of the operators during the assembly, testing and subsequent maintenance of the handling and lifting equipment.
- The warehouse is characterized by a block of shelving in two-faced single depth configuration as shown in Figure 4.2, so as to allow operators visual control of the operation of the material handling equipment, including any critical gripping and malfunctioning of it;
- A wire mesh is installed in the parts not visible and transparent protection panels in plexiglas material or scratch-proof glass are mounted in the exposed parts;
- The cells are equipped with special currents for the support and the sliding of the loading units;
- Roller conveyor systems have the upper roller wire at an altitude of between 650 and 850 mm. Furthermore, the roller transport system is sustained by vertical supporting elements with an appropriate height adjustment system with a range of 150 mm;

Different are the possible material flows across the automated warehouse. Following the numeration in Figure 4.2, the principal ones are:

- Input of totes destined to storage by AGV (9, 10, 1)
- Output of totes from rack to operator for picking operations (gravity planes):
  - Kitting workstation (1, 10, 3)
  - Picking workstation (1, 10, 4)
- Output of totes output from rack to AGV (1, 10, 2, 5)
- Output of and empty totes to kitting area for kit preparation (1, 10, 2, 6)
  - With return to rack of the newly composed tote (6, 10, 1)
  - With collection by the AGV of the newly composed tote (6, 10, 2, 5)
- Tote composition in picking area (7):
  - With pick-up of an empty tote from the exchange point with external areas (8 conveyor below)
  - With deposit of the newly composed tote on the exchange point with external areas (8 conveyor above)
  - With return to rack of the tote retrieved from rack if still containing products (9, 10, 1)
  - With transfer of the tote retrieved from rack to the exchange point if empty (4, 8)

Outside the automated area, the majority of material handling tasks is performed by 2 AGVs (see Figure 4.4). The technical characteristics of them are:

- Mechanical properties:
  - Length: 890 mm;
  - Width: 580 mm;
  - Height: 352 mm;
  - Weight (without load): 62,5 kg;
  - Loading surface: 600x800 mm;
  - Max. height from the floor with load: 1.800 mm;
  - Payload load: 100 kg (with maximum inclination 5%);
  - Payload in drag: 300 kg;
  - Operating time or duration: 10 H or 20 km;
- Speed and Performance:
  - Speed Forward: 1.5 m/sec;
  - Speed Backward: 0.3 m/sec;
  - Radius of curvature: 520 mm (around robot center);
  - Repeatability: +/- 50 mm in position - +/- 10 mm for charging;
  - Charging time: up to 3 h (0-80% 2 h);
- Sensors:
  - 2 laser scanners for 360° visual protection around the robot;
  - 1 3D camera for the detection of objects ahead the robot from 50 to 500 mm above the floor;
  - 4 ultrasonic scanners for the detection of transparent objects and glass doors.

The interface between the AGVs and the other components of the warehouse is given by motorized roller conveyors placed on the top side of the vehicle. The roller equipment installed has a built-in lifting mechanism (25 mm) that allows easy pick-up of boxes, pallets, items. The movement of the loading units is on the short side. The technical characteristics of the top conveyor are:

- Dimensions of the loading table: 800 x 600 mm;
- Conveyor area: 890 x 485 mm;
- Maximum payload: 25 kg;
- Manual or automatic height adjustment from 650 mm to 850 mm;
- Loading/unloading time: 3 s.

Figure 4.4 - AGV employed in the warehouse

The automated warehouse is provided with an information and technology system necessary for the management and handling of materials. The IT structure is illustrated in Figure 4.5.

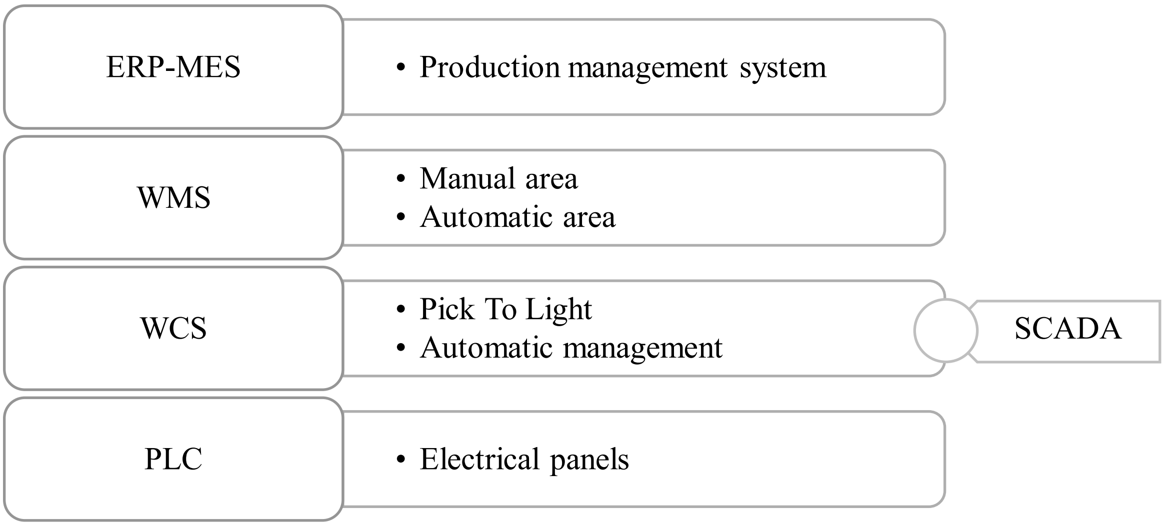


Figure 4.5 - IT structure of the automatic warehouse

The first level is composed by an ERP/MES system. ERP refers to Enterprise Resource Planning while MES stay for Manufacturing Execution System. MES is an information system used to connect and control manufacturing systems and data flows with a high level of complexity. The integration of MES in an organization aims to guarantee a correct execution of operations and enhance production output. ERP/MES level manages the planning of production needs in terms of work orders with the associated bills of materials. At each launch of the production plan, the list of bills of materials with the detail of the components and their quantities is sent to the WMS, Warehouse Management System. The single bill is activated manually when or automatically at an AGV or external event arrivals. The WMS manages the bill by dividing any manual or automatic picking missions. The latter is activated via WCS, Warehouse Control System. It is a software application which aims to coordinate daily operations within a warehouse center. The WCS represents an important connection between warehouse management software (WMS) and material handling equipment. At automatic retrieval operations, WCS sends the commands to the PLC, governing all automatic handling elements. Furthermore, through SCADA it is possible to monitor the evolution of the missions in charge.

5. The conceptual model

The second step in simulation modelling is the construction of a conceptual model. As defined in chapter 3 - Simulation modelling, a conceptual model includes all the elements that characterize the model itself, it describes the inputs and outputs of the model, it includes a description of the contents of the model regarding the environment modelled and at which degree of detail, and it finally illustrates which are the assumptions and simplifications made during the abstraction process.

In order to better explain some features of the model, the activity diagrams of the Unified Modelling Language (UML) are used, since this typology of diagram "visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modelling" (smartdraw.com).

The specifications of the project do not explain some details of the system. For this reason, it is necessary to make some assumptions about certain aspects and to hypothesize some behaviors of the processes. The aspects that are not cited in this chapter are considered as described in section 4.2 - The warehouse or considered superfluous for this study.

5.1 Layout

The specifications of the project give only some information about how the automated warehouse area is organized, but they do not consider the other processes involved in the functioning of a warehouse plant. Considering which are these procedures and which are the physical limitations of the laboratory, a layout is hypothesized. Referring to Figure 5.1, the environment is organized as follow:

1. Automated area
2. Receiving station
3. Assembly station
4. AGVs' home
5. Workers' home
6. General buffer of empty totes
7. Outbound area

The automated area presents the features described in section 4.2 - The warehouse.

The receiving station is the area dedicated to the management of external supplies and the preparation of totes for storage. It is composed by a working table and three buffers. The first buffer is the one on the left side of the bench and it is employed for the momentary storage of the boxes containing the components to be processed. The second buffer resides on the right side of the table and it is used for the empty totes which will be fulfilled with the supplied components. The last buffer is located next the first one and it is utilized to put the created totes with the components that wait to be transferred to warehouse by AGVs.

The assembly station is employed in the creation of final products through an assembly operation. It is composed by a big working table where reside some totes that will contain the created products. Furthermore, there are four buffers. The first one is on the right side of the working table and houses the totes coming from the automated area containing the components to be assembled. The second buffer stays on the left side of the table and it includes empty totes that will be used to replace the totes on the working table once they are full. The third buffer resided in front of the bench and is used to momentarily store the full totes containing products before they are picked up by AGVs and transferred to the automated area. The last buffer is employed for the totes that are emptied from the components and need to be sent to other areas of the warehouse.

AGVs' home simulates a charging point where the vehicles move when they are free of tasks. The number of vehicles is set to 2, as specified in section 4.2 - The warehouse.

Workers' home represents an area where the operators stays when they have not processes to handle. The number of workers assumed to be necessary to complete the warehouse operations amounts to 2.

The general buffer of empty totes contains a series of empty totes that can be used while necessary to perform some operations.

Finally, the outbound area simulates the space of the warehouse dedicated to all the shipment operations. In order to simplify the model, it works simply as a buffer that is emptied once reached a certain amount of ready orders.

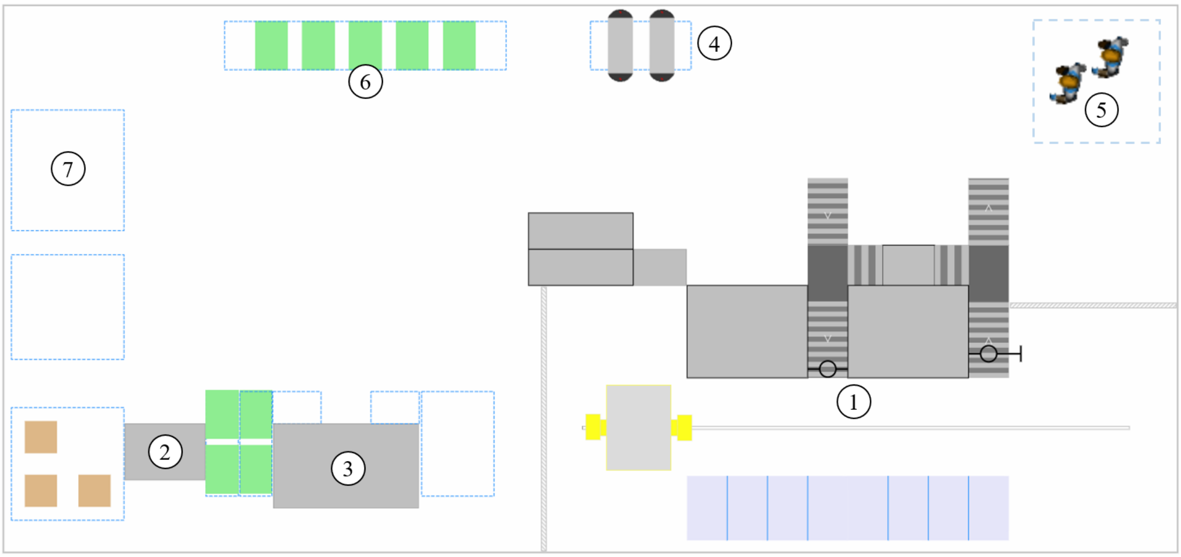


Figure 5.1 - Hypothesis of layout

5.2 *Items*

The automated warehouse handles various components contained in totes and boxes. In order to simplify the model, only a typology of container is taken into consideration, since the tote and the boxes present the same structural characteristics. The plastic totes are selected. Furthermore, to simulate the behavior of the system, three typologies of components are suggested.

The three components differentiate from each other thanks to some attributes summarized in Table 5.1:

- Type is a unique code that identifies the component;
- Color is a visual attribute that may help to evaluate the performance of the model during simulations;
- Batch size corresponds to the quantity of component supplied;
- Max tote indicates the maximum quantity of components that can be inserted in a tote;
- Order point determines the stock quantity under which a supply order is emitted.

Table 5.1 - Components' attributes

Type	Color	Batch size	Max tote	Order point
C1	Black	70	35	35
C2	Gold	90	45	45
C3	Red	80	40	40

The components are employed in the creation of kits and in the assembly of products. The proportion of components in the kits is the same needed to create a product. It is assumed the presence of three different products, that differentiate from each other thanks to some attributes summarized in Table 5.2:

- Type is a unique code that identifies the product;
- Color is a visual attribute that may help to evaluate the performance of the model during simulations;
- Number of C1 indicates the number of components C1 necessary to create the kit/assemble the product;
- Number of C2 indicates the number of components C2 necessary to create the kit/assemble the product;
- Number of C3 indicates the number of components C3 necessary to create the kit/assemble the product;

Table 5.2 - Products' attributes

Type	Color	Number of C1	Number of C2	Number of C3
P1	Orange	2	2	1
P2	Blue	3	2	4
P3	Violet	1	3	2

5.3 Mini-load stacker crane functioning and allocation logic

The handling of materials within the automated area is performed by a mini-load stacker crane. From the analysis of the specifications, there are not indications about the technical characteristics of it. Consequently, based on the mini-loads' specification of the principal brands in the market, some assumptions are made:

- Travel speed max: 6 m/s;
- Lifting speed max: 3 m/s;
- Loading/unloading fixed times: 3 s.

Since the limited areas and routes, no acceleration is taken into consideration.

Furthermore, from the project specifications it is possible to understand that the mini-load performs three main operations, that is storage, retrieval and transfer.

A storage operation consists of the pickup of a tote coming from the input conveyor and its transfer to a specific position within the rack. The mini-load starts moving from its current position horizontally and vertically simultaneously. When both the elements of the mini-load, that is the column and loading device, arrive at their relative destinations, the tote is loaded. Once completed the loading operation, a free cell is assigned to the tote, according to the allocation logic of the system. The mini-load starts moving to the target position, and once both the parts are at destination, the unloading operation is performed. After that, the inventory is updated. If the tote contains some components, the number of items is added to the stock of the specific component. If the tote is empty, it is inserted in the list of empty totes. If it contains a prepared kit, it is appended to the list of kit stored within the rack. If it contains some assembled product, the number of items is added to the stock of the specific product. Storage task may be schematized as in Figure 5.2.

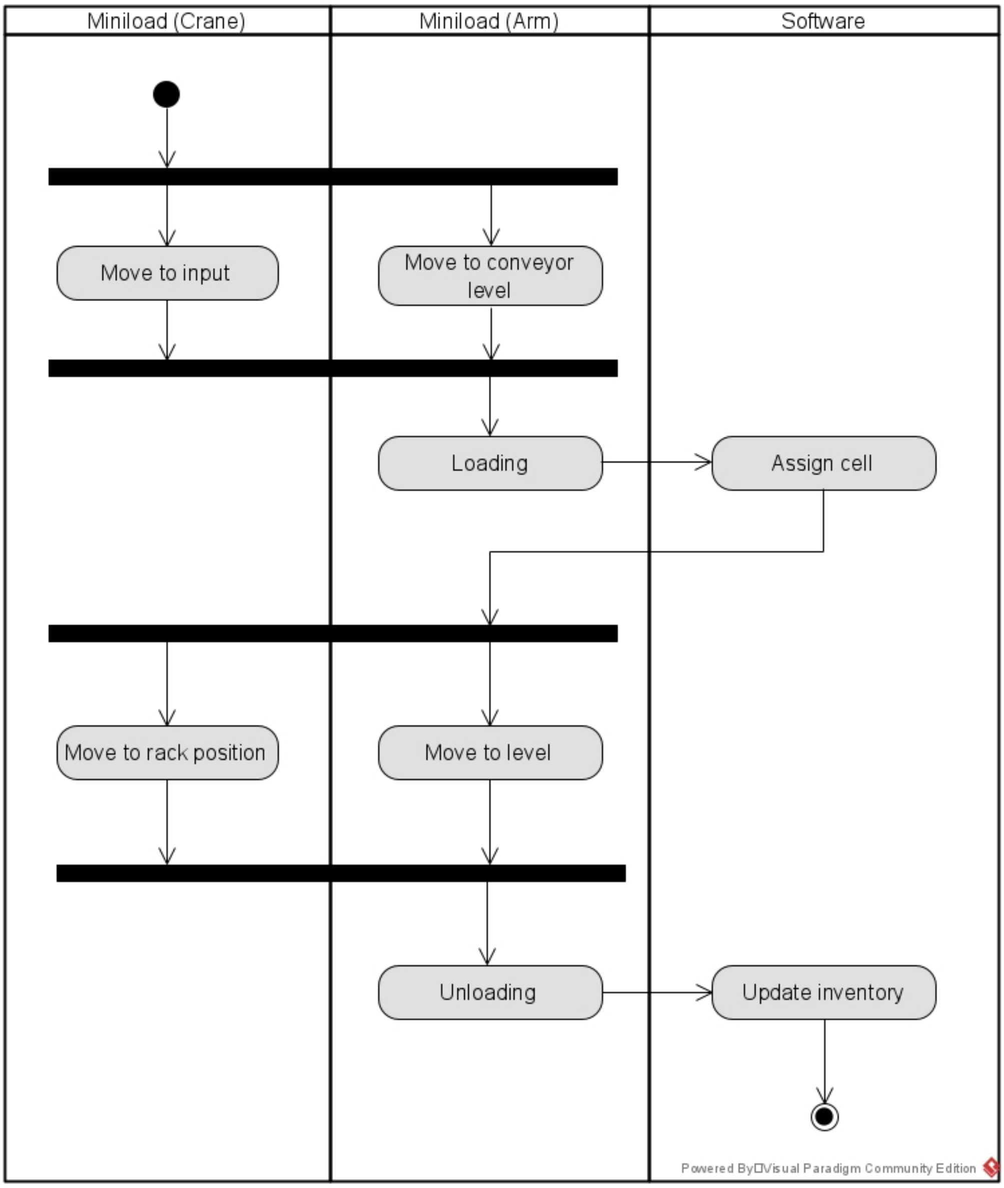


Figure 5.2 - Storage activity diagram

On the other hand, retrieval operation consists of the transfer of a tote from its rack position to a specific destination, that may be the output conveyor, the kitting station or the picking station. At generation of a retrieval task, the target position is defined and the mini-load stacker crane starts moving once received the coordinates. At destination, the loading operation is performed. Then, the column and loading device start moving to their relative destinations and once arrived the tote is unloaded. Finally, if the tote is destined to kitting or picking station, the quantity of components contained in the tote is subtracted from stock. If the tote is an empty tote, it is delated from the list of empty totes. Finally, if the tote contains an already created kit, it is cancelled from the kit stored list. The retrieval task may be schematized as depicted in Figure 5.3.

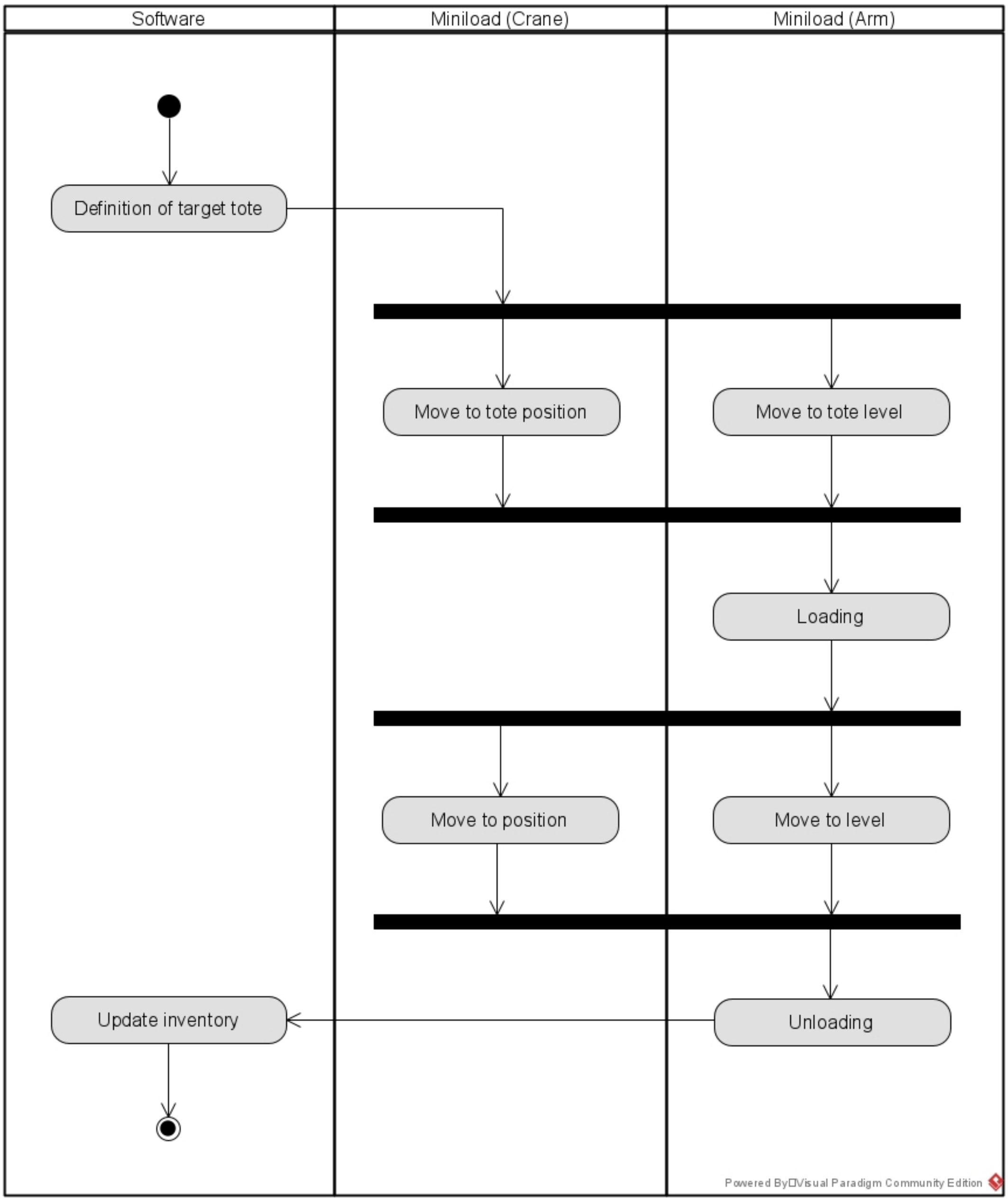


Figure 5.3 - Retrieval activity diagram

Finally, the transfer operation involves the movement of the tote from the input conveyor directly to the output conveyor. This operation is performed when a tote needs to exit automatically from the warehouse area. The transfer operation is employed in handling kits prepared in the kitting station and destined to assembly, or totes that contain a small number of components and need a refill operation. This task begins with the simultaneous movement of the column and the loading device towards the input conveyor. Successively, the tote is loaded by the loading device. The column starts moving in direction of the output conveyor, while the loading device maintains its relative position since the two conveyors have the same height. Once at position, the tote is unloaded and begins its movement towards the end of the output conveyor. The transfer operation is simplified in Figure 5.4.

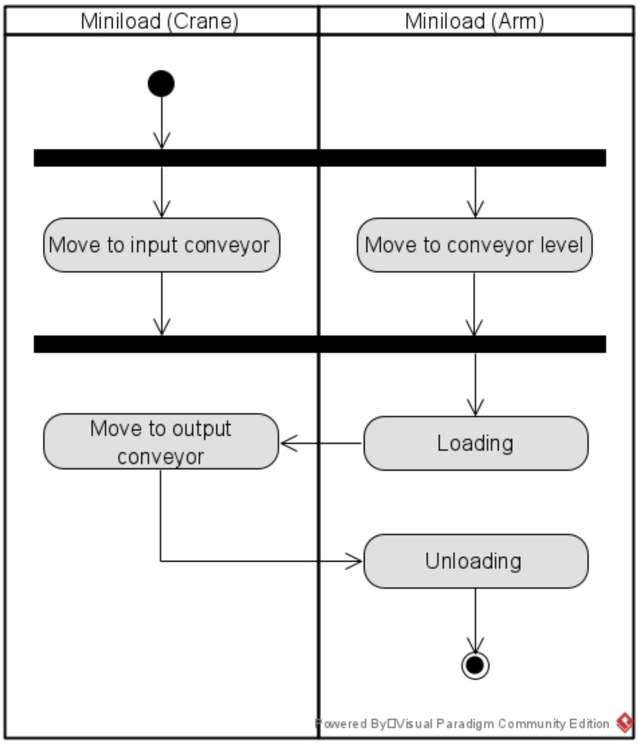


Figure 5.4 - Transfer activity diagram

Since the automated area manages empty totes, kits, components, and products, it is supposed the presence of an allocation logic to differentiate the areas of the rack in relation to the items stored. In particular, it is assumed that:

- the empty totes are stored on the side of the rack in proximity of the output conveyor;
- the completed kits are stored starting from the middle of the rack in direction of the output conveyor;
- the totes containing the components are stored starting from the middle of the rack in direction of the picking station;
- the assembled products are stored on the opposite side of the rack in relation to the empty totes, that is near the picking station.

Despite the differentiation of storage area, in every case a storage level is firstly filled completely before changing storage position, that is the totes are stored in height before moving along the rack.

5.4 Receiving process

When shipments arrive to the warehouse, the receiving process may start. From the specification analysis, this process is not taken into account and for this reason it is necessary to hypothesize its functioning. It is assumed that the receiving operations are handled in a dedicated station constituted by a working table. In order to simplify the model, no labelling, inspection and quality controls are considered. At the station, an operator picks up the components that arrive in supply boxes and put them into totes. The totes are positioned in a dedicated buffer that may include both empty totes and totes containing some components, that simply need a refill. The totes already containing components have the priority over the empty totes. The quantity of components in each tote is defined by the attribute Max tote of the Table 5.1. The created totes are moved in a dedicated buffer waiting to be transferred by AGVs to the automated area. Once picked up all the components, the supply box is removed from the environment. The receiving process is summarized in Figure 5.5.

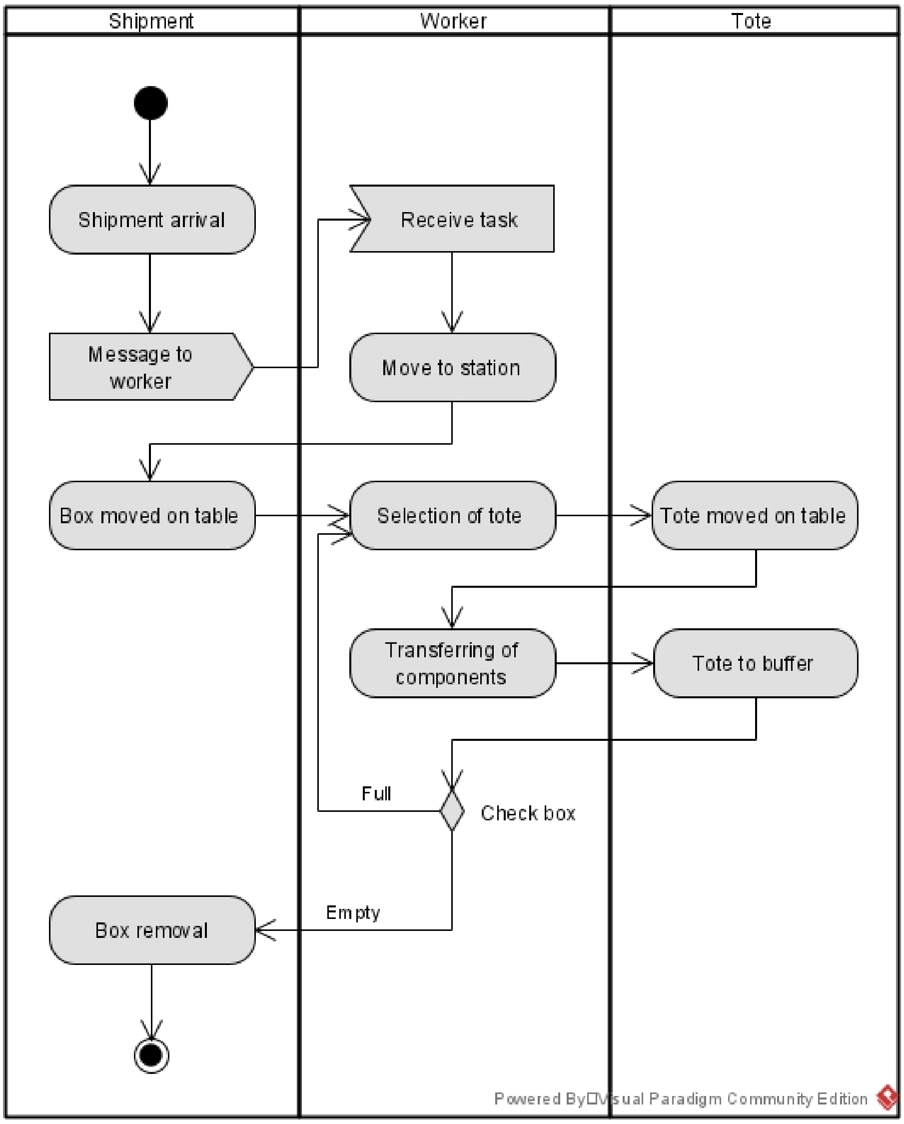


Figure 5.5 - Receiving activity diagram

5.5 *Put-away process*

Once the components are received from supply and inserted in the totes, they are ready to be stored. In order to be allocated into a position within the rack, the tote has to be transferred from the receiving station to the automated warehouse. Like the case of receiving, also this process has to be suggested since no indications are given in the specifications of the project. Once positioned in the dedicated buffer, the management software of the warehouse checks if in the rack there is enough space for another tote, also considering possible totes already transferring to the automated area. If the control is negative, the put-away process ends, and it is delayed until the space condition becomes true. If it is positive, a message is sent to the AGVs fleet. The first free vehicle moves in direction of the buffer, loads the tote and heads to the input conveyor of the automated warehouse. Once at position, the tote is unloaded, and it is transferred by the conveyor to the interface point with the mini-load stacker crane. A message to the stacker crane is send which is ready to perform a storage task. The put-away process is summarized in Figure 5.6.

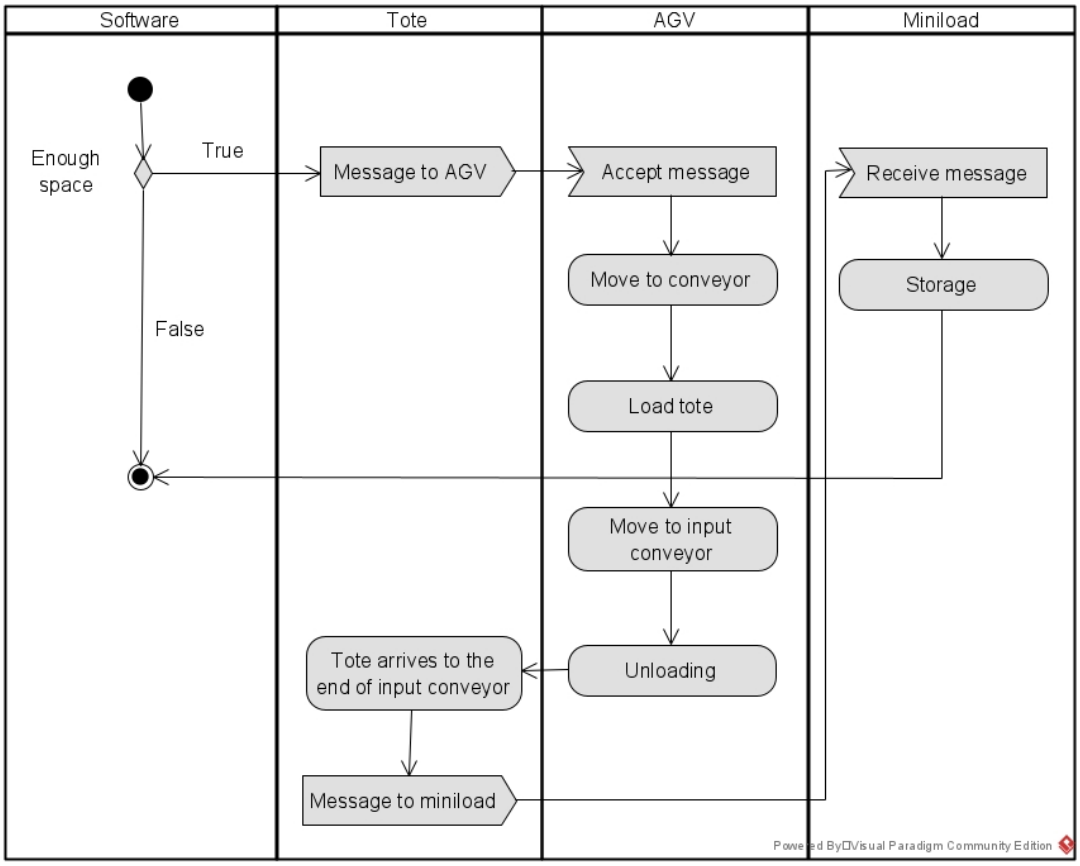


Figure 5.6 – Put-away activity diagram

5.6 Kitting process

The kitting station is employed in the creation of kits through a pick-to-light system that facilitate the operator in the performance of the task. The kits created in this working station may destined to storage or intended for assembly. At kitting order arrival, the management software executes a control about the completion of a previous order, that is it checks if a worker is processing another order at kitting station. Additionally, it controls if the stock is completed, that is if there is at least one tote per component stored within the rack. If the control is negative, the new order is added to a queue. On the other hand, if the check is positive, the order is accepted and a message to retrieve an empty tote is sent to the mini-load stacker crane. Then the software controls if in the gravity planes is stored at least one tote for each typology of component. If yes, the task of the mini-load stacker crane ends, if not it retrieves the missing tote or totes in order to refill the gravity planes. Meanwhile, at arrival of the empty tote at the station, a message is sent to the worker, which accepts the task and heads to the station. The worker waits until completion of the mini-load task and subsequently creates the kit. Once the kit is completed, the tote is moved to the conveyor and transferred to the rack. Furthermore, if the totes stored in the gravity planes contain a small number of components after creation of the kit, they are removed from their position and loaded on conveyor. If their content is equal to 0, they are sent to rack as empty totes, while if the content is greater than 0, they are transfer by the mini-load stacker crane and they are send to the receiving station for a refill operation. The completion of a kitting order is exemplified in Figure 5.7.

The operator spends a certain amount of time to complete the operation. Not having real data regarding the operational times, it is necessary to estimate the duration of the operation. The hypothetical execution time is obtained listing the activities that the operator makes and applying the MTM-UAS method to calculate the time spent to complete the operation. MTM-UAS was born as an aggregate method to describe elementary actions in the field of batch production and the automotive industry. Since 2004, it has become a generally accepted method of time analysis and it is applied to situations where work cycles have a duration of the minute. This methodology uses alpha-numerical codes descriptive of the elementary actions and to which are associated specific time values useful to calculate the total duration of the work cycle. The value found through this methodology is used to determine the probabilistic distribution of execution times of a kitting operation.

The type of distribution selected is the triangular one. The triangular distribution is a continuous probability distribution and it is so called since it is shaped like a triangle. It becomes useful to determine the distribution when limited sample data is available. It is characterized by three parameters, that is a lower limit, a peak value and an upper limit. The value found through the MTM-UAS method is used as lower limit, while the peak value corresponds to the lower limit increased by 20%, and the upper limit is equal the lower limit augmented by 40%. Consequently, the lower limit is 9.5 s, the peak value is 11.5 s, while the upper limit is 14 s.

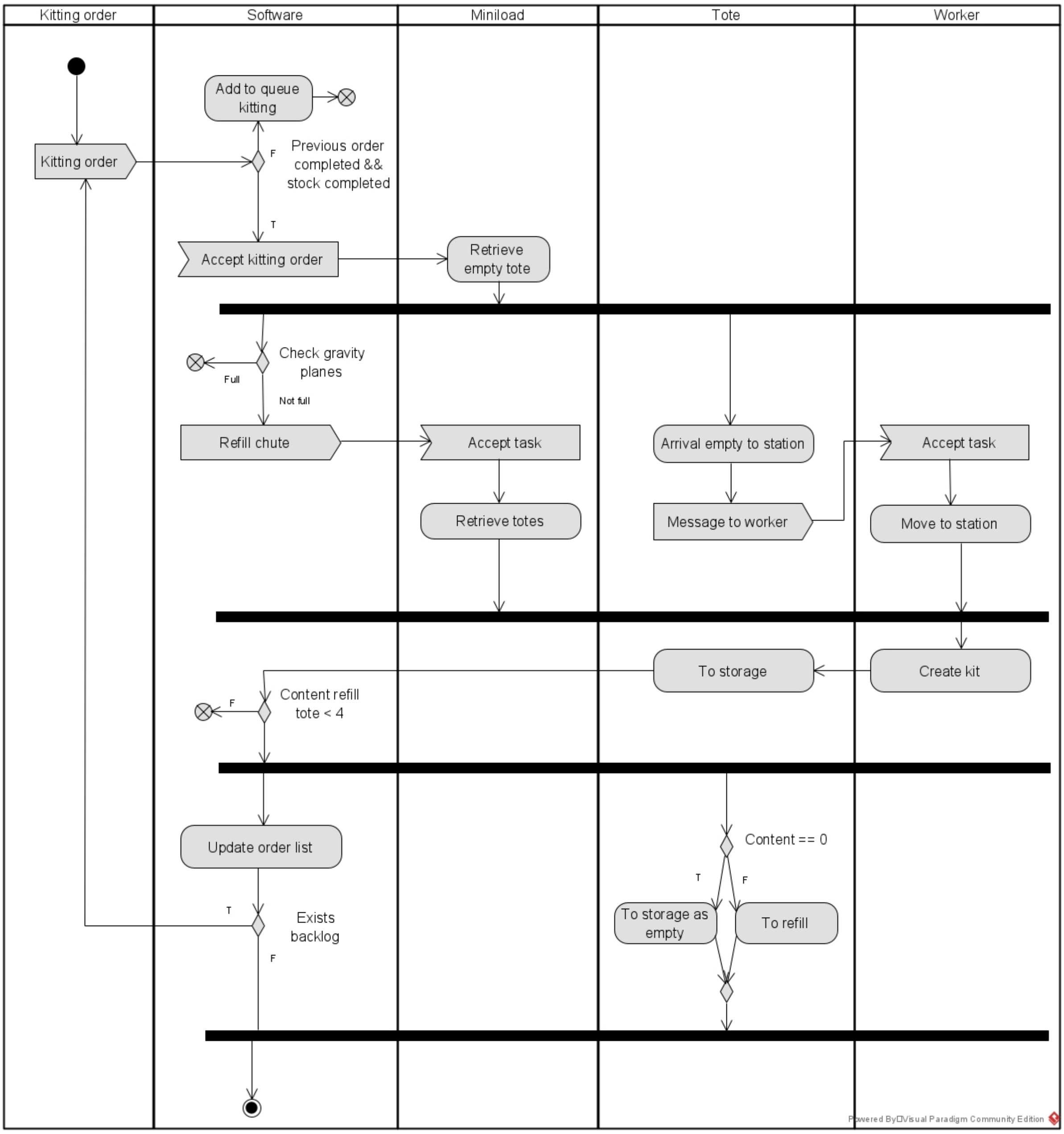


Figure 5.7 - Kitting activity diagram

5.7 Assembly process

The assembly process consists of the creation of a final product putting together a series of components in specific quantities (see Table 5.2). This process is not explicated into the project specifications and for this reason it is necessary to hypothesize its functioning. It is assumed that the assembly operations are carried out in a dedicated station near the automated area. An assembly order may be generated automatically by the system on a rate basis, or specially if there are not enough products stored within the rack to complete the picking orders. The completion of an assembly order is exemplified in Figure 5.8.

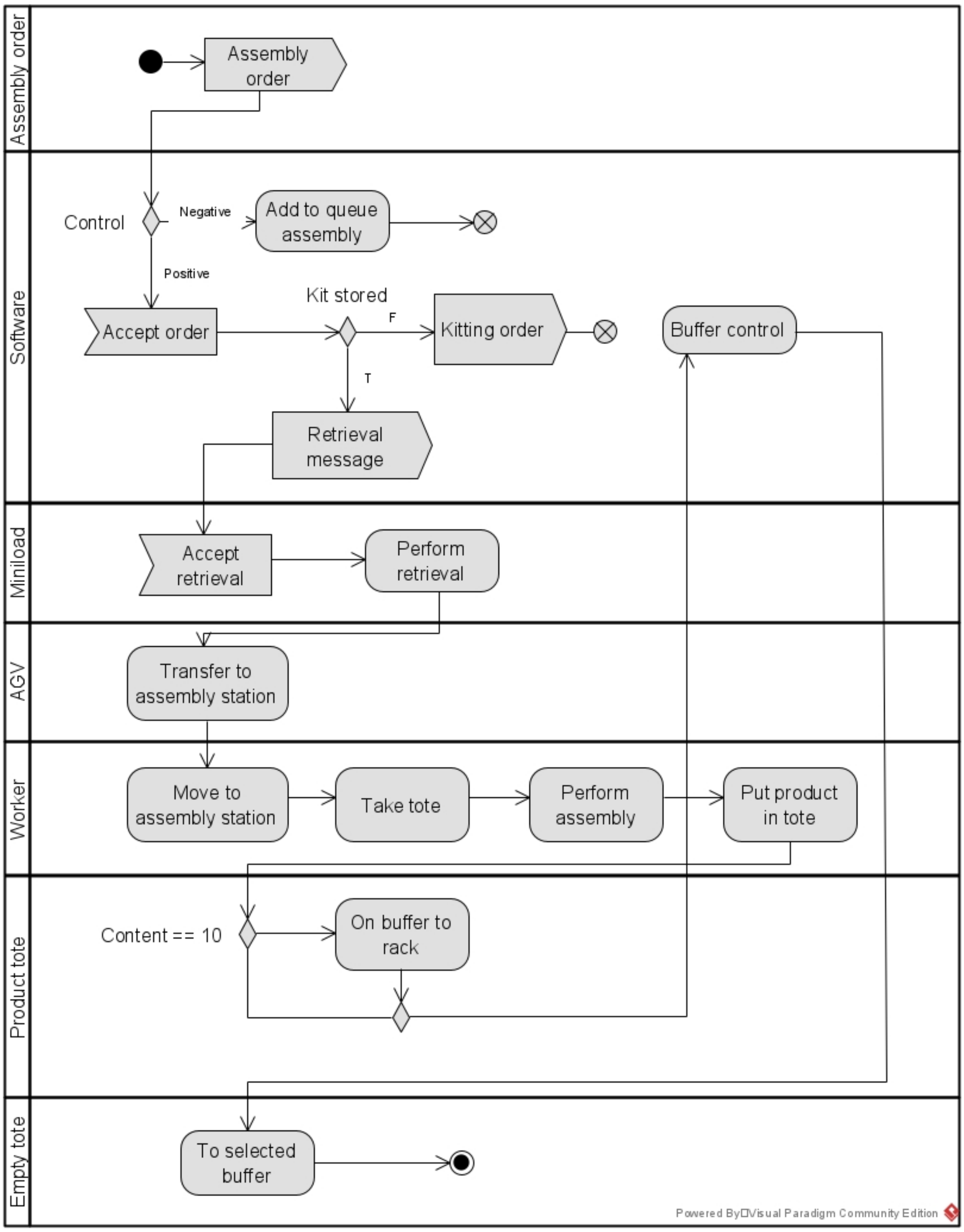


Figure 5.8 - Assembly activity diagram

At arrival of the assembly order, the software checks the possibility to complete the order. In particular, it controls if there is enough space at assembly station buffer to hold another tote. If the control is positive, the software accepts the assembly order, while if the check is negative, it adds the order to queue. Successively, the software monitors the presence of at least a kit stored in the rack of the typology requested in the order. In case of availability of a kit, the software sends a retrieval message to the mini-load stacker crane for the kit found. If there is not a kit stored, a kitting order is sent to the system. In this case, instead of being stored within the rack, the tote containing the kit is transferred out by the mini-load stacker crane and directly sent to the assembly station.

When the tote containing the kit arrives at the end of the output conveyor, a message is sent to the AGVs fleet. The first free vehicle approaches to the conveyor, it loads the tote and it moves in direction of the assembly station. When arrived at destination, the AGV unloads the tote in a dedicated buffer next to the station. The operator assigned to this process moves to position, he takes the tote containing the kit, he picks up all the components contained in it and he builds the product. The product is then inserted in dedicated totes, which are sent to rack once their content is equal to 10. These totes contain the products necessary to successively complete the picking orders. After the assembly, the tote that contained the kit is empty and it is left near the assembly station if the number of empty totes necessary to insert the created products is lower than 4, it is moved manually to the buffer of empty totes of the picking station if it is not full, it is moved manually to the buffer of empty totes near the receiving station if its content is lower than 20, it is transferred by AGV to the rack if the number of empty totes stored is lower than 6 or it is relocated to the general buffer of empty totes in the other cases.

The time needed to complete an assembly operation is estimated subjectively, without following a precise method like in the case of kitting process. This is due the difficulty to state a plausible value unknowing the complexity of the process. For this reason, it is assumed a triangular distribution, with lower limit equal to 180 s, peak value equivalent to 240 s, while the upper limit amounting to 300 s.

5.8 Picking process

In addition to kitting and assembly operations, also picking activities are performed in the warehouse. The picking process occurs in the picking station. A worker is in charge to create a customer order getting the quantity of products requested from a tote and create another tote destined to shipping.

At arrival of a picking order, the software monitors the presence of at least a tote stored in the rack containing the products of the typology requested in the order. In case of availability of a tote, the software sends a retrieval message to the mini-load stacker crane for the tote found. If there is not a tote stored, an assembly order is sent to the system. In case of availability, the mini-load stacker crane accepts the retrieval task and the tote to get is selected. Then, it moves to target position and it retrieves the tote, transferring it to the picking station. Once the retrieval operation is completed, an operator moves to the station to create the customer order. At position, he takes an empty tote from the buffer, he positions it on the table near the picking station, he gets the number of components from the tote moved by the mini-load and he inserts them in the empty tote. Once completed the order, the tote is moved on the exchange conveyor with the external areas. Finally, if the tote containing the products to be picked is empty, it is moved to the buffer, while if it contains some items, it is reinserted to rack. The picking process is schematized in Figure 5.9.

The method to estimate the time needed to complete all the picking operation is the same followed for the kitting process. Consequently, the lower limit corresponds to 11.5 s, the peak value is 14 s, while the upper limit amounts to 17 s.

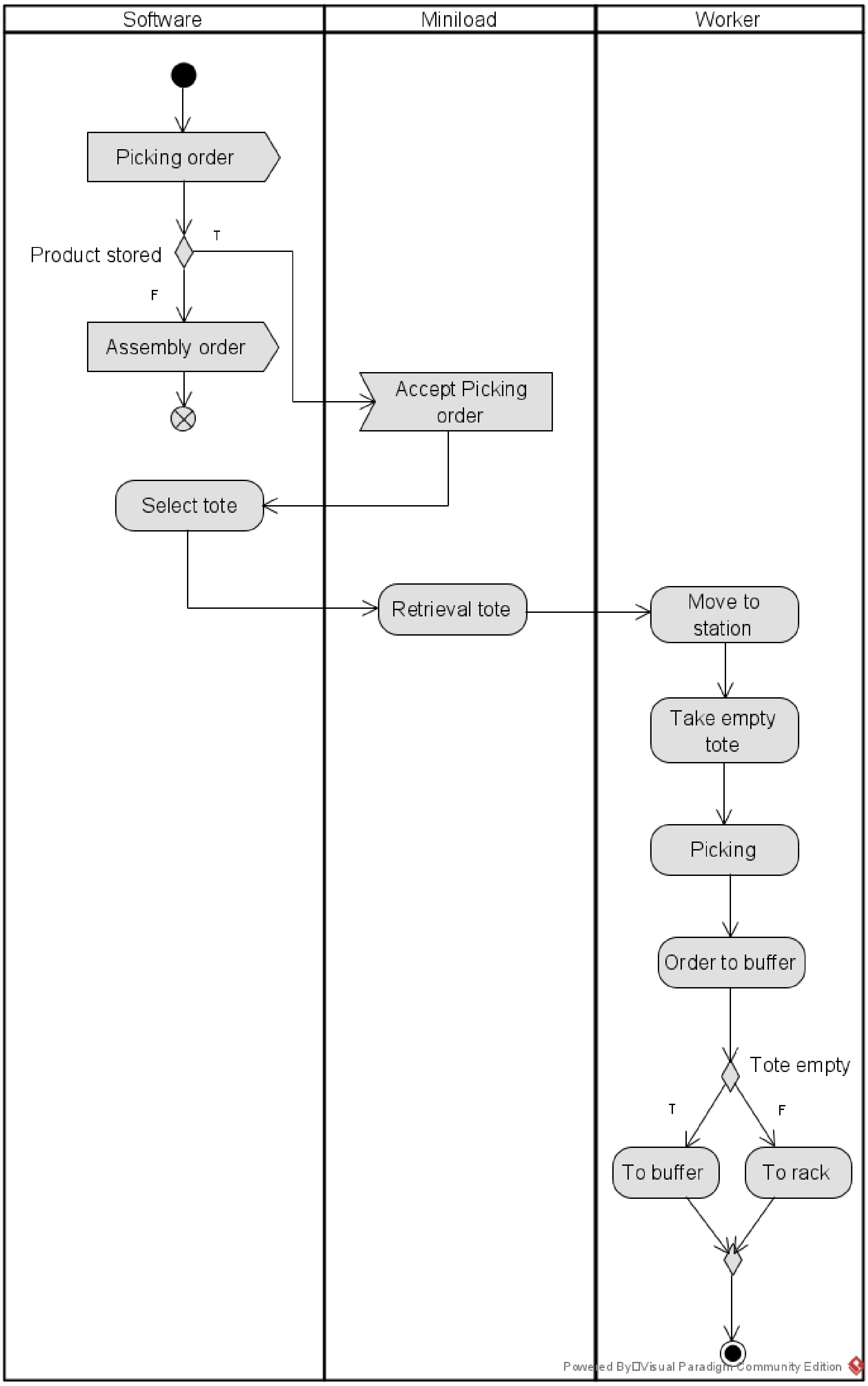


Figure 5.9 - Picking activity diagram

5.9 Outbound

The case study into analysis does not consider outbound operations like packaging and shipment. Nevertheless, once the customer orders are prepared in the picking station, they are moved to a buffer that simulates the outbound area of the warehouse. In order to simplify the model, it is emptied once reached a certain amount of ready orders. This aspect of a warehouse is not deepened as the project is thought with research perspective and does not include the actual delivery of products.

5.10 Stock and material flow management

Important aspect to take into consideration is the stock and flow management. Stock management is a set of procedures, policies and controls that monitors the quantities in stock and determines which level to maintain, when to reinstate and which dimensions must have the orders. To simplify the inventory management system, it is assumed an order point for the components, that is a level of stock under which it is emitted a supply order by the system. The order quantities are fixed (see Table 5.1). On the other hand, the stock management of products operates in a different way. A part of the stock of products is handled through a push strategy, that is assembly orders are generated on a rate basis. The other part is controlled through a pull strategy, that is the assembly orders are created by the picking process when it is impossible to complete a picking order due to an absence of stock.

Material flow management consists of directing and controlling the sequence of activities from the supply of raw materials to the distribution of final products. Material flow management is introduced within the model in order to balance material flows and to prevent any block of the system. It controls:

- The availability of space within the rack to guarantee the possibility to stock another tote;
- The free capacity of conveyors to value the chance to transport totes.

Without material flow management, the system would fall into errors like collision of totes, block of the mini-load stacker crane, or congestion of flows.

5.11 Inputs and outputs

As defined in chapter 3 - Simulation modelling, the inputs of a conceptual model are the elements that are modified throughout simulation modelling in order to attain the objectives identified. The input identified are three parameters:

1. Kitting rate;
2. Assembly rate;
3. Picking rate.

The kitting rate correspond to the rate at which kitting orders are generated. This rate does not include the internal kitting order created to complete an assembly order when there is not an already prepared kit stored in the rack. The assembly rate represents the rate at which assembly orders are produced. It does not include the internal assembly order automatically generated once product stock is insufficient to complete a picking order. Finally, picking rate indicates the rate at which picking orders arrives at warehouse.

Since the objectives of this simulation model are the evaluation of the system performance and the identification of an optimal design solution to be implemented, a series of experiments are identified. Each experiment differs from another thanks to changes in input parameters or other aspects.

Four main experiments are identified:

1. Base model;
2. Demand rates;
3. Layout changes;
4. Priority changes.

The first experiment refers to the simulations on the model as it is built and described in this chapter. The second experiment considers the variation of the demand rates, in order to understand which is the performance of the warehouse in different demand situations. The third experiment involves some layout changes of the elements outside the automated area, i.e. the receiving station, the AGVs' home, the assembly station, and the various buffers. The last one includes the variation of the processes' priority in order to value the reaction of the system to such changes. All the experiment are examined in depth in chapter 7 - Experiments and results.

The results of the experiments consist of the output of the model, that is statistics which apprise us if modelling purposes have been reached and otherwise the reasons behind the unexpected results. The key performance indicators identified are of two typologies, that is time indicators (Table 5.3) and productivity indicators (Table 5.4). Since the absence of real data from the warehouse, quality and price KPIs are not considered.

Table 5.3 - Time indicators

Time indicator	Meaning
Storage cycle time	Mean time taken during the entire process of each storage task
Retrieval cycle time	Mean time taken during the entire process of each retrieval task
Transfer cycle time	Mean time taken during the entire process of each transfer task
Mini-load cycle time	Mean time taken during the entire process of a mini-load task
Mini-load waiting time	Mean time waiting for a mini-load operation
Kitting waiting time	Mean time between kitting order arrival and its taking charge
Kitting transferring time	Mean time between kitting order taking charge and its processing
Kitting processing time	Mean time between kitting order processing and its conclusion
Kitting cycle time	Kitting transferring time + Kitting processing time
Kitting order time	Kitting waiting time + Kitting cycle time
Assembly waiting time	Mean time between assembly order arrival and its taking charge

Time indicator	Meaning
Assembly transferring time	Mean time between assembly order taking charge and its processing
Assembly processing time	Mean time between assembly order processing and its conclusion
Assembly cycle time	Assembly transferring time + Assembly processing time
Assembly order time	Assembly waiting time + Assembly cycle time
Picking waiting time	Mean time between picking order arrival and its taking charge
Picking transferring time	Mean time between picking order taking charge and its processing
Picking processing time	Mean time between picking order processing and its conclusion
Picking cycle time	Picking transferring time + Picking processing time
Picking order time	Picking waiting time + Picking cycle time

Table 5.4 - Productivity indicators

Productivity indicator	Meaning
AGVs utilization	Percentage of time in which the AGVs fleet is occupied
Operator 1 utilization	Percentage of time in which the operator 1 is occupied
Operator 2 utilization	Percentage of time in which the operator 2 is occupied
Mini-load utilization	Percentage of time in which the mini-load stacker crane is occupied
Receiving utilization	Percentage of time in which the receiving station is occupied
Kitting utilization	Percentage of time in which the kitting station is occupied
Assembly utilization	Percentage of time in which the assembly station is occupied
Picking utilization	Percentage of time in which the picking station is occupied
Rack utilization	Percentage of space occupied within the rack
Mini-load throughput	Operation per hour
Kitting throughput	Completed kitting order per hour
Assembly throughput	Completed assembly order per hour
Picking throughput	Completed picking order per hour
Mini-load WIP	Average mini-load WIP
Kitting WIP	Average kitting station WIP
Assembly WIP	Average assembly station WIP
Picking WIP	Average picking station WIP

5.12 Other aspects

A real system presents failures and breakdown of the machineries and would require maintenance operations to maximize the availability of the equipment. Due the absence of real data of failures necessary to estimate the time distribution of breakdowns and consequent maintenance, no maintenance and breakdowns are introduced within the model.

Furthermore, the speed of conveyors is set at a mean value that amounts to 0.5 m/s, while the switching delay of transfer tables is set to 1.5 s.

The operators walking speed is assumed to be that of an average walk, that is 1.25 m/s.

6. The virtual model

The following step in simulation modelling consists of translating the conceptual model into a virtual model. Between conceptual model and virtual model there would be another step called model design. As defined in chapter 3 - Simulation modelling, it consists of the structured creation of objects for the computer model. In this study, the model design and virtual model constitute a single phase of simulation modelling. The passage to a virtual model involves the translation of the conceptual model into a specific computer code. The environment selected to execute this step is the simulation software AnyLogic.

6.1 The simulation software: AnyLogic

AnyLogic is a multi-method simulation software born in 2000. It is one of the products provided by The AnyLogic Company, a multinational team that operates from US and Europe with a global network of partners. It offers AnyLogic, a general-purpose simulation software, AnyLogic Cloud, a cloud platform destined to model execution and integration, and anyLogistix, a supply chain design optimization and innovation software. AnyLogic is the flagship of The AnyLogic Company and it is used in many businesses and academic institutions, including over 40% of Fortune 100 companies. This software is employed in various industries, comprising supply chains, manufacturing, transportation, warehouse operations, rail logistics, mining, oil & gas, ports & terminal, road traffic, passenger terminals, healthcare, business processes, asset management, marketing, social processes and defense.

AnyLogic provides a set of libraries and industry-specific tools that permits modelling at different levels of detail. The instruments used in the creation of the model of the warehouse are:

- Space markup elements;
- Process modelling library;
- Material handling library;
- Additional agent’s elements;
- Data analysis instruments.

6.1.1 Space markup elements

To model the elements of the warehouse, a series of constructs embedded in the software called space markup elements is employed. They permit to characterize the environment of the model and to define agent locations. Furthermore, the space markup elements allow to animate the agents that live in the model environment. Various are the objects that could be modelled, but the most used in warehouse simulation are described in Table 6.1.

Table 6.1 - AnyLogic Space Markup elements (help.anylogic.com)

Icon	Space markup	Description
	Path	Path graphically defines a movement path for agents. Nodes can be connected with paths. Altogether they compose a network. In the network, paths connecting nodes define the routes that agents may take when moving from one node to another
	Rectangular Node	Node defines a place where agents can reside. Nodes can be connected with paths. Altogether they compose a network. In the network, node defines a place where agents may reside
	Point Node	Point node usually defines a transit transportation node in a network. It is a node that does not have area but has just a point size
	Attractor	Attractor allows controlling agent's location inside a rectangular node or a polygonal node. If the node defines the destination of the agent movement, attractors define exact positions inside the node. If the node defines the waiting location, attractors define exact points where agents will wait inside the node. Agents will go to attractor location for waiting
	Pallet Rack	The space markup element Pallet Rack graphically defines the pallet rack used in warehouses and storage zones
	Conveyor	Conveyor is the space markup shape that graphically defines a conveyor
	Position on Conveyor	Position on conveyor is the graphical element that is used to define the exact position on the conveyor. It can be used to define the location where new material items will be placed on the conveyor, to set the destination point for the material items being transported by conveyor(s), to simulate photo eyes, scanners and other devices that perform some instant actions with the conveyed material items, to model different types of stops and escapement
	Transfer Table	Transfer table is the space markup element that is used to define transfer tables in the material handling models. Agents (material items) passing through it keep their current orientation in space
	Custom Station	Custom station is the space markup element used in material handling models. It defines a station/working zone where material items are processed. The process is not set up in this element

6.1.2 Process modelling library

Space markup elements are not sufficient to completely model the warehouse since they only allow to design the physical aspect of the system. In order to configure the process in terms of agents, operations and resources, AnyLogic offers the process modeling library which permits the creation of process-centric models of real-world systems. This library contains many blocks that enable the configuration of the processes in form of flowcharts. AnyLogic flowcharts are hierarchical, scalable, and extensible. Furthermore, they are object oriented since the software is based on Java programming language. Thanks to these characteristics, the software allows modelling of complex systems at any level of detail.

Another important feature to consider about the process modelling library is the possibility to animate the processes designed through flowcharts. The animation is defined linking a space markup element to a process block. Then while agents reside inside that block, the agent animation shapes are represented at the corresponding positions of the space markup element.

Table 6.2 illustrates the most used process modelling library blocks during the development of the warehouse model.

Table 6.2 - AnyLogic process modelling library (help.anylogic.com)

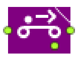
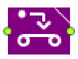
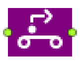
Icon	Block name	Description
	Source	Generates agents
	Sink	Disposes incoming agents
	Delay	Delays agents by the specified delay time
	Queue	Stores agents in the specified order
	Wait	This block is like Queue block with one exception: it supports manual retrieval. It has no ordering (except the case when pre-emption occurs, if the latter is turned on)
	Select Output	Forwards the agent to one of the output ports depending on the condition
	Hold	Blocks/unblocks the agent flow
	Move To	Moves an agent from its current location to new location
	Resource Pool	Provides resource units that are seized and released by agents
	Seize	Seizes the number of units of the specified resource required by the agent
	Release	Releases resource units previously seized by the agent
	Enter	Inserts agents created elsewhere into the flowchart
	Exit	Accepts incoming agents
	Batch	Accumulates agents, then outputs them contained in a new agent
	Unbatch	Extracts all agents contained in the incoming agent and outputs them

Icon	Block name	Description
	Dropoff	Extracts the selected agents from the contents of the incoming agent
	Pickup	Adds the selected agents to the contents of the incoming agent
	Resource Attach	Attaches a set of portable and/or moving resources to the agent
	Resource Detach	Detaches previously attached resources from the agent
	Rack System	Models a storage zone containing a set of racks (defined by PalletRack shapes), providing centralized access and managing of racks
	Rack Store	Places an agent into a cell of the specified rack (PalletRack) or storage zone (RackSystem)
	Select Output In	Both with SelectOutputOut acts as two halves of large multi-exit SelectOutput block
	Select Output Out	Both with SelectOutputIn acts as two halves of large multi-exit SelectOutput block

6.1.3 Material Handling Library

In addition to Process Modelling Library, AnyLogic offers another series of block usable in material handling cases. Among all, the ones employees in the model are shown in Table 6.3.

Table 6.3 - AnyLogic Material Handling library (help.anylogic.com)

Icon	Block Name	Description
	Convey	Transports the incoming agents by conveyor(s) to the specified destination point. It is the one and only block that controls material items movement within a conveyor network
	Conveyor Enter	Places the incoming agents in a conveyor network but does not start the items transportation by a conveyor
	Conveyor Exit	Removes the incoming material items from a conveyor network and sends them further via the output port as regular agents
	Transporter Fleet	Defines a fleet of transporters used in material handling process. An example of a transporter is an AGV (automated guided vehicle)
	Move by Transporter	Performs transportation of an agent by a transporter. The block provides a complete set of parameters for seizing a transporter, loading it with the required agent, sending it to the specified location, unloading the agent there, and finally releasing the transporter
	Seize Transporter	Seizes one transporter from the specified fleet defined by the TransporterFleet block. Sends the seized transporter to the specified location.

6.1.4 Additional agent's elements

Based on the characteristics of agent-based modelling paradigm, a model is composed by different agents with different characteristics and properties that acting together generate the simulation and abstraction of the real-world problem. In order to describe the features of the agents within the model it is possible to insert other elements in addition to the ones described above. These elements are analyzed in Table 6.4.

Table 6.4 - Additional agent's elements (help.anylogic.com)

Element	**Description**
Parameters	Parameters are generally constants representing particular characteristics of an object within the model. They become fundamental in the description of different instances of the same agent which differ precisely in the value assumed by the parameter
Variables	Variables represent model state and may change during the simulation. They are normally used to model some dynamic object characteristics or to store the results of model simulation. Variables are values of a Java class or some arbitrary scalar type
Collections	A group of objects of the same class generates a collection. Collections allow to define this group of elements into a single unit. For this reason, they permit to manipulate aggregated data. Various are the type of collections usable in AnyLogic like array list, linked list, linked hash set, tree set, tree map, linked hash map. These differ for the methodology of access and manipulation of data. In the model proposed only the first kind was used. In particular, array lists consist of resizable array provided by its own capacity. When an element in added to the array list, capacity increases automatically
Option List	Option lists are an embedded element of the software that is used in order to define some agent attribute that can assume only particular alternative values or option. Each option list is composed by specific elements which represent the different occurrences that the attribute may assume. Once defined, option list becomes a type assignable to parameters or variables
Functions	AnyLogic permits the definition of own functions. They prove to be useful when it is needed to carry out some operation that is difficult to model directly with the objects and elements embedded in the software. Another important potentiality of functions is the possibility to use them in different places and different times in the model. Functions are able to return specific values as result of the same or simply run the code and execute the action. They are written in Java and it is possible to take advantage of all the peculiarity of this language
Events	Events are the instrument to schedule some actions within the model. They are mainly used when is needed to repeat cyclically an operation. Timeout triggered and cyclic events are used within the model. After a specified moment, the event occurs on the basis of a particular rate

It is possible that agents present some more sophisticated behavior that is not possible to define with the simple use of the elements above mentioned. In these cases, AnyLogic offers another modelling instrument called state charts. State charts are a relevant construct that might be useful to describe events and time driven activities. With their introduction within the model, it is possible to animate a larger series of discrete events than the ones reproducible with block-based elements. An example of state chart can be consulted at Figure 6.11. They are constituted by states and transition. States represent the particular condition of the agent in a specific moment during simulation. Transitions consist of the passage from one state to another one. These ones may be caused by different conditions. When the state chart assumes a particular state, all possible triggers that may cause the transition to another state are collected and the chart waits the occurrence of one of these triggering events or conditions. Transition may be produced by a series of event and the ones used in model are described in Table 6.5.

Table 6.5 - Trigger types in state charts (source: help.anylogic.com)

Type	**Description**
Timeout	Transition to the following state happens only at timeout expiration, counted from the instant the state chart enters the state that precedes such timeout transition. The timeout expression can be stochastic or deterministic
Message	Transition occurs upon reception of a specific message received by the state chart or by the agent from outside
Agent arrival	Transition takes place when the agent reaches the destination point of the movement initiated with specific functions into the state that precedes the transition

6.1.5 Data analysis instruments

AnyLogic also offers some tools to collect statistics regarding the model and analyze them, such as datasets and charts. Dataset allows the storage of 2D (X, Y) data of type double. Furthermore, it is capable of calculate and maintain updated the minimum and maximum values of the stored data for each dimension. The dimension of the dataset is limited, and it only preserves the indicated number of the latest data values. This means that adding a new element to a full dataset will cause the loss of the oldest value. The double dimension of the dataset allows to use time as X value and observe the modification of the Y value along simulation, or it is possible to store the dependency of one value on another.

In addition to datasets, AnyLogic includes a set of chats that can be employed to graphically represent both simulation output data and simulation runtime data. The elements and features of the charts can be edited in AnyLogic environment. This tool allows to link charts to parameters, variables and datasets and display them at runtime. Data items can be added or deleted from the graphs dynamically. Various are the typologies of charts offered by AnyLogic, but the one employed in this model are bar charts. A bar chart shows a series of data items in form of bars aligned at one end. The sizes of them are proportional to the related data items values. If the value is negative, bars can grow in the opposite direction. An example of bar chart is shown in Figure 6.30.

6.2 *Structure of the model*

AnyLogic models are distinguished by a hierarchical structure. Each agent may encapsulate other agents at various level of depth. This structure generates a tree of agents with different ramifications. The highest-level agent is called top-level agent. It corresponds to the roots of the tree of agents and represents the highest level of abstraction of the model. Any possible agent included in the top-level agent generate a lower level of abstraction. This peculiarity allows to build a model at any level of detail desired and eventually hide a particular complexity of an object. In addition, it confers a good level of flexibility to modelling in terms of structure of the model and nature of agents.

The model is composed by one top-level agent called Main and 10 lower-level agents. The 10 lower-level agents are:

1. Component;
2. Product;
3. Tote;
4. Box;
5. AGV;
6. Worker;
7. Task;
8. Arm;
9. Crane;
10. Order.

In turn, each agent listed above contain a series of agents represented by the process blocks described in Table 6.2 and in Table 6.3.

The hierarchical structure of the model analyzed assumes the shape showed in Figure 6.1.

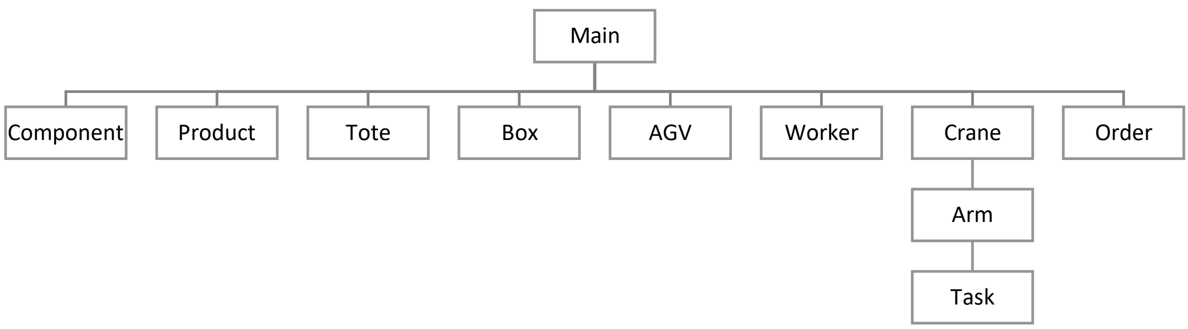


Figure 6.1 - Hierarchy tree of agents in the model

Before starting the analysis of agents, it is important to understand how the model loading works. Once the top-level agent is constructed with the set-up of all the parameters and initialization of variables, the animation is created. Each object living in workspace calls a method (name of a Java function) which creates and initializes its state charts and events. After this step, the simulation can start.

In order to better analyze the structure of the model, a bottom-up approach will be followed. So, lower-level agents will be firstly described and then it will go up along the hierarchical tree structure.

6.3 Lower-level agents

6.3.1 Component

To represent the components handled in the warehouse, the agent Component is introduced within the model. The three different types of components described in Table 5.1 do not constitute three different agents, but three instances of the same agent. This modelling choice was made due to the partial differences between the components. In fact, the variations do not extend to state charts, flowcharts, visualization or other relevant elements, but they are limited to the agent's attributes.

Only one parameter characterizes the agent Component and permits the differentiation between the three instances of the agent. It is called type and it can assume only the three values defined through the option list ComponentType. It may assume the following values:

- C1;
- C2;
- C3.

The other attributes cited in Table 5.1 are written in AnyLogic in a table called components_sheet. It is possible to extrapolate them through SQL queries when needed.

Within the model, the component is represented by a small cube with measure of 10 x 10 x 10 cm, as depicted in Figure 6.2. In order to better differentiate the components during simulation, a color is associated to a type of component, as defined in Table 5.1.

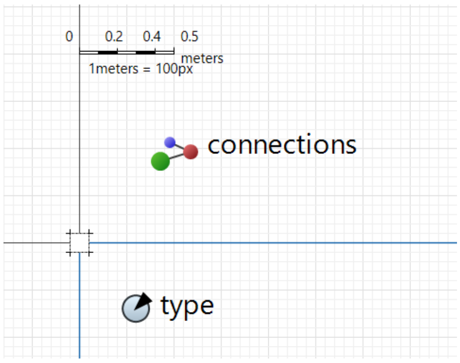


Figure 6.2 - Agent Component in AnyLogic environment

6.3.2 Product

To represent the product assembled in assembly station, the agent Product is modelled. Similarly to agent Component, agent Product may be present in three different instances constituent the three type of products that can be assembled. Exactly like in case of components, agent Product is defined by the unique parameter type which differentiate the instances of the agent. It can assume only the three values defined through the option list ProductType. It may assume the following values:

- P1;
- P2;
- P3.

Other attributes cited in Table 5.2 are written in AnyLogic in a table called products_sheet. It is possible to extrapolate them through SQL queries when needed.

Within the model, the products are represented by a cylinder with 10 cm radius and 20 cm height, as depicted in Figure 6.3. In order to better differentiate the products during simulation, a color is associated to a type of product, as defined in Table 5.2.

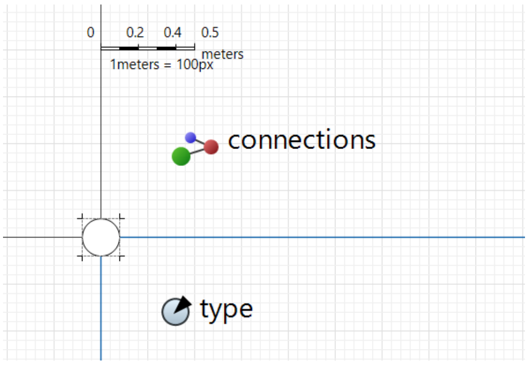


Figure 6.3 - Agent Product in AnyLogic environment

6.3.3 Tote

Within the warehouse, items are collected in plastic totes that measure 400 X 600 X 175 mm. They are modelled as the agent Tote. It is represented by a parallelepiped with the mentioned measures. During simulation, if the tote is empty the shape assumes the green color, while if it contains some element it takes the color of the object inside it. In Figure 6.4 is possible to observe the agent presentation in AnyLogic.

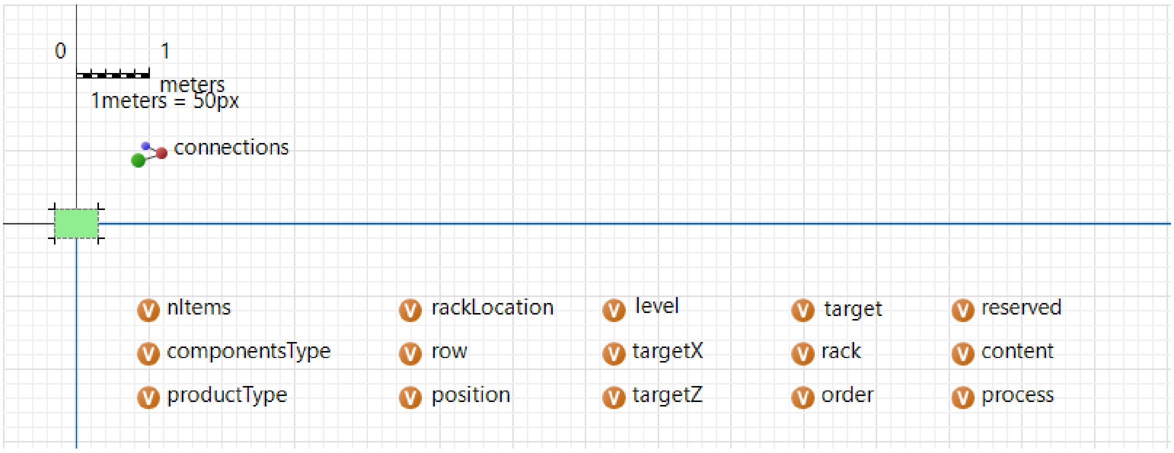


Figure 6.4 - Agent Tote in AnyLogic environment

In order to define all the characteristics of the instances of the agent, numerous are the variables introduced. These are described in Table 6.6.

Table 6.6 - Description of agent Tote's variables

Name	**Element**	**Java type**	**Description**
nItems	Variable	Integer	Number of items contained in tote
componentsType	Variable	ComponentType	Type of components contained in tote
productType	Variable	ProductType	Type of products contained in tote
process	Variable	Processes	Type of process assigned to a specific tote
rackLocation	Variable	PalletRackLocation	Java type to define the location of tote within the rack
row	Variable	Integer	Row of the tote within the rack system
position	Variable	Integer	Position of the tote within the row
level	Variable	Integer	Level of storage of the tote within the rack
target	Variable	Position	It contains the target position of the tote for storage and retrieval operations performed by mini-load stacker crane
targetX	Variable	Double	Value of the longitudinal target position for stacker crane movement
targetZ	Variable	Double	Value of the vertical target position for stacker crane movement
rack	Variable	Integer	Defines the number of the rack in which the tote is stored
reserved	Variable	Boolean	Defines if the tote is already reserved for an operation
content	Variable	ContentType	Defines the category of items that are contained in the tote
order	Variable	Order	Defines the order assigned to a tote

The variable process is defined by the variable type Processes. It is generated by an option list called Processes containing all the possible processes that may be assigned to a tote. Values contained in this option list are:

- Kitting;
- Chute;
- Picking;
- Assembly;
- Shipping;
- Storage;
- Refill.

The variable content is defined by the variable type ContentType. It is generated by an option list called ContentType containing all the possible categories of item that may be contained in a tote. Values contained in this option list are:

- KitToStore;
- KitToAssemble;
- Empty;
- Components;
- Products.

6.3.4 Box

Agent Box is modelled in order to contain the components coming from supply (see Table 5.1). This agent enters the model when a supply order is launched. Agent Box is defined by only one parameter called componentsType of type ComponentsType, that determines the type of components contained in the box when it arrives to the warehouse.

In AnyLogic, Box is represented by a 50x50x50 cm light brown cube, as shown in Figure 6.5.

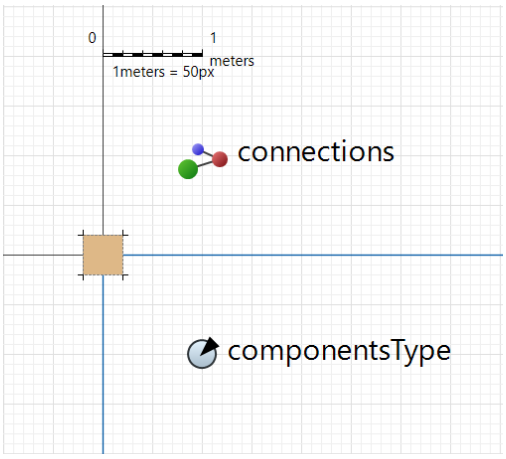


Figure 6.5 - Agent Box in AnyLogic environment

6.3.5 AGV

Agent AGV is used within the model to handle empty and full totes. It is not described by any parameters or variable since the instances of the agent are all equal. It is characterized by all the mechanical and physical characteristics as defined in project specifications. It is treated as transporter within the model.

In AnyLogic, agent AGV is represented as shown in Figure 6.6.

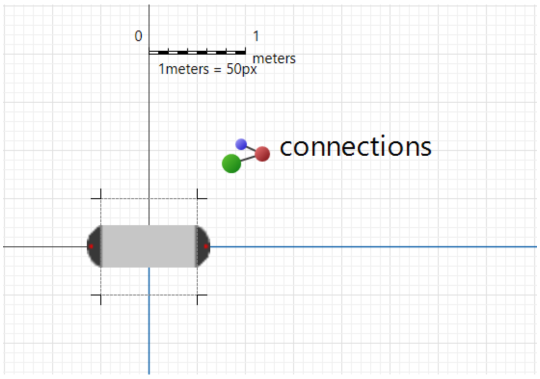


Figure 6.6 - Agent AGV in AnyLogic environment

6.3.6 Worker

Agent Worker is introduced with the aim of carrying out some of the activities present in the warehouse like receiving, kitting, assembly and picking processes. It is not described by any parameters or variable since the instances of the agent are all equal. It constitutes a resource within the model.

In Figure 6.7 is possible to see agent presentation in AnyLogic.

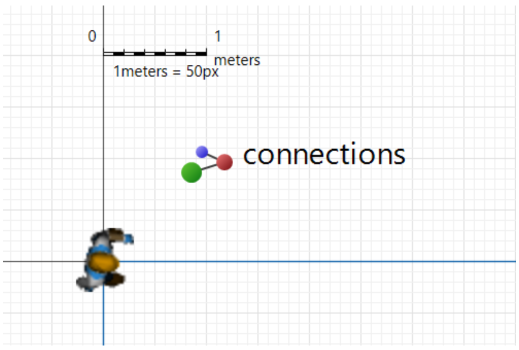


Figure 6.7 - Agent Worker in AnyLogic environment

6.3.7 Task

Task is an agent introduced into the model to better manage the behavior of the mini-load stacker crane. In fact, it is used to launch a storage, retrieval or transfer job. Unlike the other agents, Task does not present a physical aspect, since it is used similarly to a message. Differently from a simple message send between agents, Task possesses all the properties of an agent, so it can be generated or destroyed, it is delayable and it can be stopped. Each instance of the agent is defined by a series of parameters described in Table 6.7.

Table 6.7 - Description of agent Task's parameters

Name	**Element**	**Java type**	**Description**
storage	Parameter	Boolean	Defines if the task is a storage task
retrieval	Parameter	Boolean	Defines if the task is a retrieval task
transfer	Parameter	Boolean	Defines if the task is a transfer task
priority	Parameter	Integer	Determines the priority of the task
type	Parameter	TaskType	Defines the target of the task

The parameter type is defined by the Java type TaskType. It is generated by an option list called TaskType containing all the possible targets of the mini-load stacker crane. It is needed to launch specific Main functions in relation to the value of the parameters itself (see sub-section 6.3.9 - Crane). Values contained in this option list are:

- taskStorage;
- taskPicking;
- taskKitting;
- taskEmpty;
- taskEmpty1;
- taskAssembly.

In Figure 6.8 is shown agent presentation in AnyLogic environment.

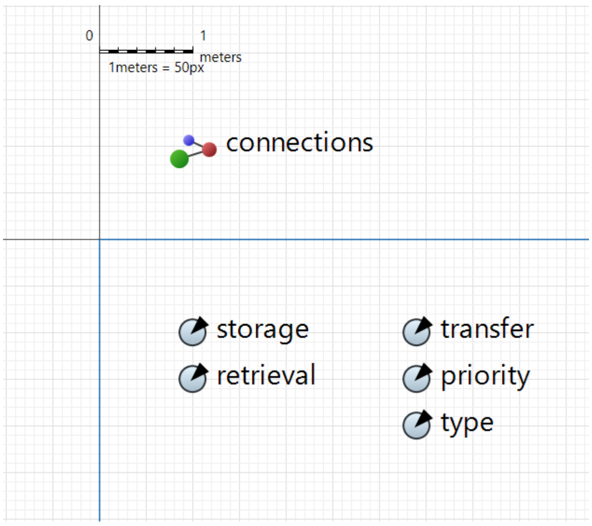


Figure 6.8 - Agent Task in AnyLogic environment

6.3.8 Arm

Within the model, the two elements that constitute the mini-load stacker crane, that it the column and the lifting device, are represented as two separate agents. This modeling choice is made to better manage the behavior of the two items. In fact, agent Arm consists of the load lifting device connected to column of the mini-load stacker crane. The agent Arm is one level lower than the agent Crane, which represents the stacker crane as a whole (see Figure 6.1 and sub-section 9 - Crane;). It introduced within the model to manage the vertical movement of the stacker crane.

Arm is characterized by a series of parameters and variables, described in Table 6.8.

Table 6.8 - Description of agent Arm's elements

Name	**Element**	**Java type**	**Description**
zConveyor	Parameter	Double	The value of height of input and output conveyors
zKitting	Parameter	Double	The value of height of the kitting station gravity planes
zPicking	Parameter	Double	The value of height of the picking station gravity planes
linkToCrane	Parameter	Crane	This parameter permits link of Arm with the upper-level agent Crane
verticalSpeed	Parameter	Double	The value of vertical speed of the load lifting device
boxStorage	Variable	Tote	The instance of agent Tote destined to storage
boxRetrieval	Variable	Tote	The instance of agent Tote destined to retrieval
boxTransfer	Variable	Tote	The instance of agent Tote destined to transfer
currentTask	Variable	Task	The current instance of agent Task performed by Arm
currentTaskCrane	Variable	Task	The current instance of agent Task performed by Crane

In Figure 6.9 is shown the agent presentation in AnyLogic.

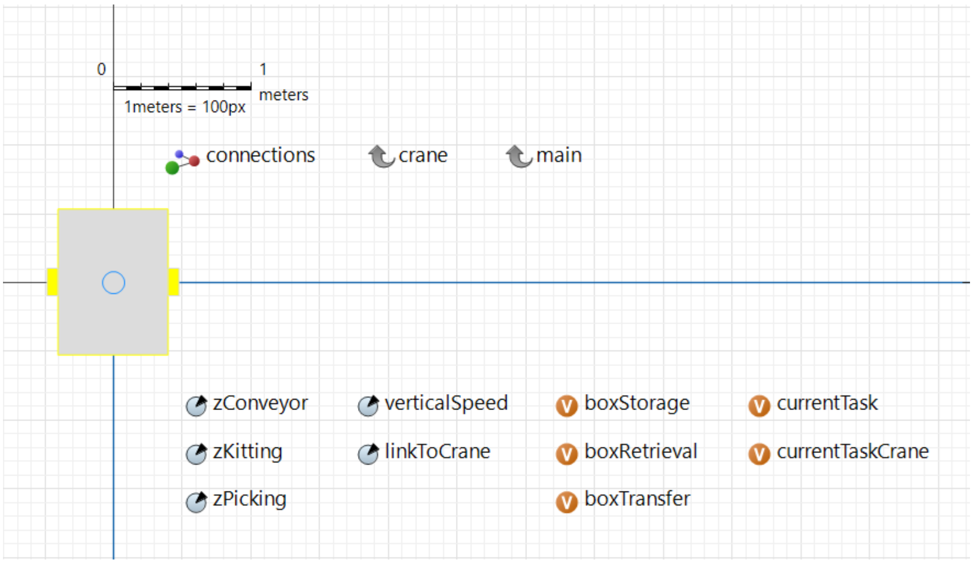


Figure 6.9 - Agent Arm in AnyLogic environment

To determine the behavior of this agent a flowchart is built (see Figure 6.10).

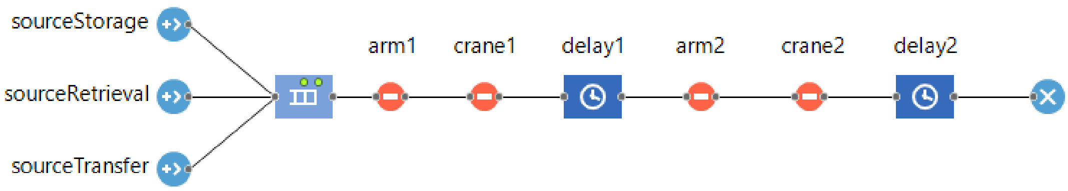


Figure 6.10 – Arm's flowchart

At occurrence, the three sources generate respectively a storage, retrieval and transfer task. The agent Task created enters the flow chart and it is stored in a priority-based queue. The priority according to which operations are selected is the one that determines the Task instance. Despite the presence of three typologies of tasks that can be carried out by the Arm, the behavioral logic is always the same. In fact, the Arm moves to a target position, it loads a tote, it moves to a second target position and it unloads the tote. The target positions depend on the task. The queue is able to store multiple agents, but only one at the time can exit from it when both arm1 and crane1 are unblocked. The two mentioned hold blocks turn into unblock state when respectively Arm and Crane reach their first target position. Once arm1 and crane1 are unblocked, the current Task enters the block delay1, which delays the agent for the necessary time to perform the loading operation. Once terminated this action, the agent enters the second part of the flowchart. Once the Arm and the Crane are at their relative second target position, arm 2 and crane 2 turn into unblocked and the agent Task can be delayed for the time necessary to unload the tote. When the process is completely ended, Task enters a sink and it is cancelled from workspace. At this point, another task if present can flow along the chart.

The flowchart above described is necessary to manage the lifecycle of each Task assigned to the mini-load stacker crane. Precisely for this reason, agent Arm in not able perform any operation only with it. To enable Arm to move and handle the totes the state chart depicted in Figure 6.11 is introduced within the model.

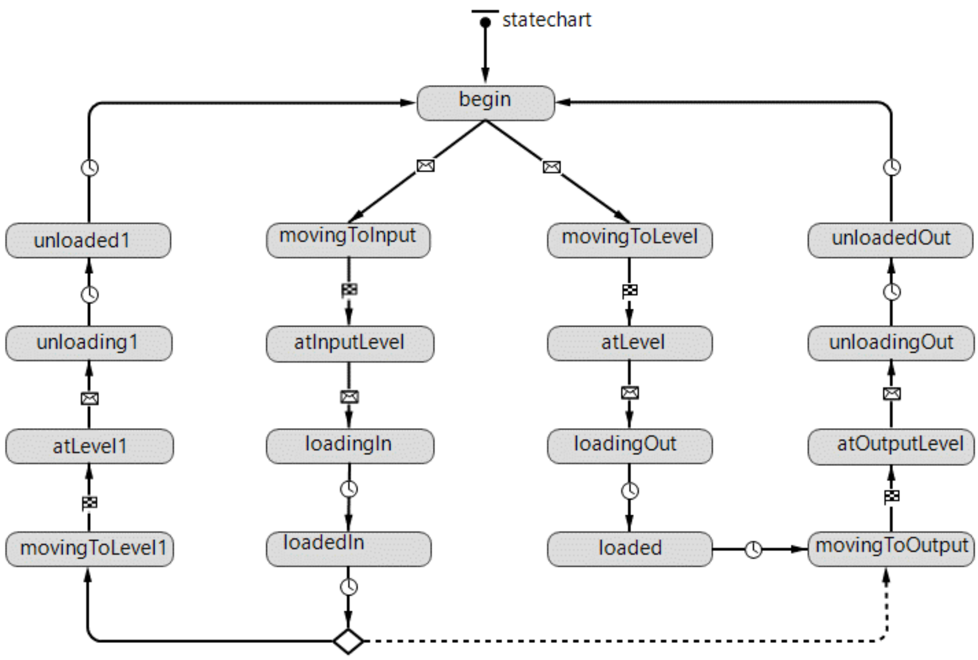


Figure 6.11 - Arm's state chart

From initial state begin, the state chart can move along two possible paths, the left one dedicated to storage tasks and the right one reserved to retrieval tasks. State chart enters into the correct branch depending on the message received by flowchart at task generation (Figure 6.10). If a storage message is sent to state chart, Arm enters movingToInput state which determines the translation of the agent to input position (zConveyor parameter). When arm reaches the specific coordinate, transition is triggered, and state chart passes to the following state. To receive the message that the system is ready to load the tote, i.e. arm and crane are located at input position, loadinIn state is activated. After 4 seconds of loading operation, Arm is able to move to the storage level assigned to the carried tote. When the agent arrives at position, state chart enters atLevel1 state. Exactly like before, when both arm and crane agents have completed the movement, another message is sent to state chart, which can enter unloading1 state. The storage operation totally lasts 4 seconds, at which end the performed task is concluded and it exits flowchart (Figure 6.10).

On the other hand, if a retrieval message is received, state chart enters the right branch of the state chart. Arm starts its movement in direction to the rack level of the target retrieval box. When level is reached, state chart is able to enter atLevel state and waits the message that confirms the readiness to load the box. Totes need 4 seconds to be retrieved from rack. After that, arm starts moving in direction of the output z coordinate, which corresponds to the height of the target destination, depending on the process assigned to the retrieved tote. When both arm and crane are located at correct position, a message is sent to state chart which enters unloadingOut state. The operation takes 4 seconds to be performed and on receipt of the unloading completed message, state chart returns to begin state and retrieval task can be considered concluded (Figure 6.10).

If Arm has to perform a transfer task, the state chart moves along the left path. In fact, the first part of this operation is the same of a storage task, that is the approach to the input conveyor. After having loaded the tote, the state chart exits the left branch and moves to the right one. This happens at the branch after loadedIn state. Indeed, the second part of a transfer task corresponds to the second phase of a retrieval operation, since the tote has to be transported to output conveyor.

6.3.9 Crane

Crane is the agent that reproduces the mini-load stacker crane as a whole. Crane is a higher-level agent compared to Arm, to the extent that it includes this last agent. Indeed, Crane illustration is composed by Arm representation and two rectangular elements which act as the columns of the mini-load stacker crane. The hierarchy between the two agents is also observable from Crane representation in AnyLogic environment (see Figure 6.12). This agent is introduced within the model to manage the horizontal movement of the stacker crane. It is characterized by a series of parameters, variables and one collection, analyzed in Table 6.9.

Table 6.9 - Description of agent Crane's elements

Name	Element	Java type	Description
homeX	Parameter	Double	The absolute x coordinate of the starting position of mini-load stacker crane
homeY	Parameter	Double	The absolute y coordinate of the starting position of mini-load stacker crane
horizontalSpeed	Parameter	Double	The value of horizontal speed of the mini-load stacker crane
xAGVInput	Parameter	Double	The absolute x coordinate of the input conveyor
xOutput	Parameter	Double	The absolute x coordinate of the output conveyor
xPicking	Parameter	Double	The absolute x coordinate of the picking station
xKitting	Parameter	Double	The absolute x coordinate of the kitting station
boxStorage	Variable	Tote	The instance of agent Tote destined to storage
boxRetrieval	Variable	Tote	The instance of agent Tote destined to retrieval
boxTransfer	Variable	Tote	The instance of agent Tote destined to transfer
currentTask	Variable	Task	The current instance of agent Task performed by Crane
currentTaskArm	Variable	Task	The current instance of agent Task performed by Arm

Name	Element	Java type	Description
storageOps	Variable	Integer	The number of storage operations concluded
retrievalOps	Variable	Integer	The number of retrieval operations concluded
transferOps	Variable	Integer	The number of transfer operations concluded
retrievalList	Collection	Tote	List of the totes to be retrieved

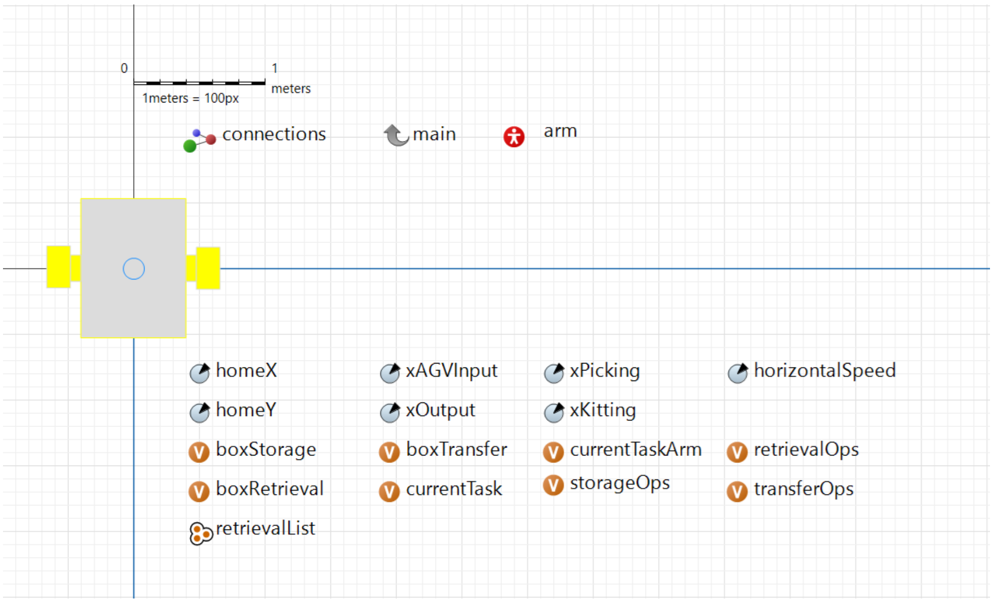


Figure 6.12 - Agent Crane in AnyLogic environment

The communication of the tasks to Crane is similar to the case of Arm. Indeed, it is introduced the flowchart represented in Figure 6.13, which is basically identical to the one in Figure 6.10, as well as its functioning. Three are the main differences. The first is the presence of a hold block after enterRetrieval that needs to stop the injection of a retrieval task if the output conveyor is full and cannot hold another tote. The second are the time measurement blocks that are necessary to calculate the queue and execution time of a mini-load operation. The third one is embedded in the queue block, that is a reference of some Main functions necessary to select a target for the mini-load stacker crane. The execution of these functions occurs once the agent exits the queue. In relation of the type of task that has to be performed, a specific function is launched (see Table 6.12 and see Attachment I - Functions):

- taskPicking calls selectPickingBox
- taskEmpty calls selectEmptyBox(kitToAssemble)
- traskEmpty1 calls selectEmptyBox(kitToStore)
- taskAssembly calls selectStoredKit
- Other type values call selectKittingBox

After the completion of the function called, the mini-load stacker crane has the information about the target to be reached and can start performing the task assigned.

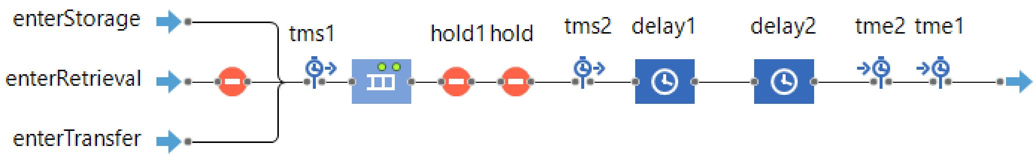


Figure 6.13 - Crane's flowchart 1

As well as the Arm case, the flowchart in Figure 6.13 does not allow the movement of the agent, since it refers to the lifecycle of a task. For this reason, a state chart is added to the model (see Figure 6.14).

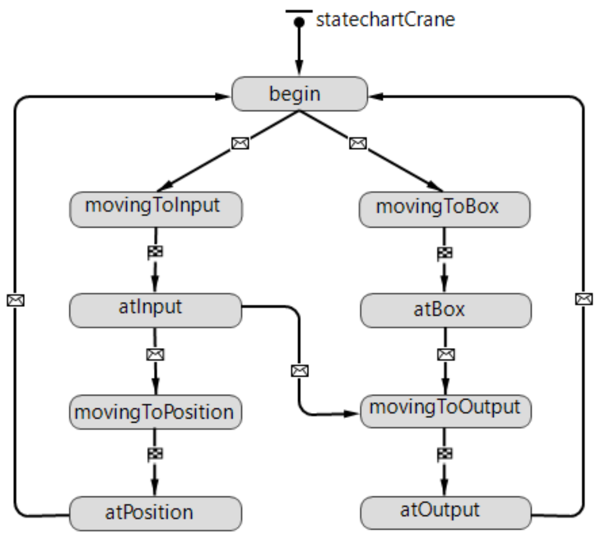


Figure 6.14 - Crane's state chart

From initial state begin, the state chart can move along two possible paths: the left one dedicated to storage tasks, while the right one reserved to retrieval tasks. State chart enters into the correct branch depending on the message received at task generation by flowchart (Figure 6.14). If a storage message is sent to state chart, Crane enters movingToInput state which determines the translation of the agent to input position (xAGVInput parameter). When Crane reaches the specific coordinate, transition is triggered, and state chart passes to the following state. When the Arm has correctly loaded the tote, a message is sent to the state chart that can enter into the state movingToPosition. The transition to the following state is triggered by agent arrival at target position. Once the tote is unloaded by Arm, a message is sent to the state chart that can return to the begin state.

On the other hand, if a retrieval message is received, state chart enters the right branch. Crane starts its movement in direction to the target horizontal position. When the position is reached, state chart is able to enter atBox state and waits the message from Arm that confirms the effectuated loading. After that, Crane enters the movingToOutput state and starts moving in direction of the final destination target x coordinate. At arrival the chart enters the state atOutput and at loading occurrence by Arm, a message triggers the transition to the begin state.

If Crane has to perform a transfer task, the state chart moves along the left path. In fact, the first part of this operation is the same of a storage task, that is the approach to the input conveyor.

After loading completion, the state chart exits the left branch and moves to the right one. Indeed, the second part of a transfer task corresponds to the second phase of a retrieval operation, since the tote has to be transported to output conveyor.

Additionally, agent Crane is characterized by another flowchart, as depicted in Figure 6.15. This flowchart is included in the model to permit to agent Tote to exit agent Main, to drop down into a lower level of the hierarchal tree and to enter into agent Crane. From the port fromConveyor, the totes that needs to be stored or transferred out enter agent Crane. After that, they have to be delayed for the time necessary to complete the storage or transfer task in delays waitingStorage or waitingTransfer. On the other hand, from the port fromRack, the totes that are retrieved enter agent Crane. Then, agents Tote enter the delay waitingRetrieval. All the three delays have not a fixed delay time, but they are interrupted only once the relative task is completed. The agent Tote returns to Main once it exits from the delays.

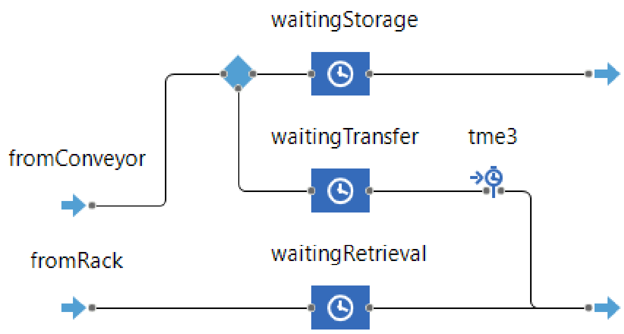


Figure 6.15 - Crane's flowchart 2

6.3.10 Order

Agent Order is introduced within the model to represent the internal and external orders that can determine the workflow of the warehouse. As well as agent Task, it does not have a physical representation, but it functions as a message or information. It is characterized by a series of parameters and variables, as summarized in Table 6.10.

Table 6.10 - Description of agent Order's elements

Name	Element	Java type	Description
type	Parameter	Processes	The type of order. It may assume the kitting, assembly and picking value
product	Parameter	ProductType	The product to be kitted, assembled or picked
timeOfArrival	Variable	Double	The time of the arrival
timeOfTransferring	Variable	Double	The time of the taking-over
timeOfProcessing	Variable	Double	The time of starting processing
timeOfCompletion	Variable	Double	The time of completion

The four variables are employed in the calculation of the statistics descripted in sub-section 5.11 - Inputs and outputs.

Order's lifecycle is determined by a state chart (see Figure 6.16). At order arrival, the state chart enters the waiting state and the instance of arrival is saved in the variable timeOfArrival. When the order is taken over and it is transferred to the dedicated station, a message triggers the transition and the state chart moves to transferring state. At this point, the instance of state passage is saved in timeOfTransferring variable. When it starts to be processed, the instance is kept in timeOfProcessing variable and a transition of state occurs. At process completion, a message triggers the completion state and the instance of the event is recorded in timeOfCompletion variable. After 1 additional second, the state chart reaches the exit point and the order is eliminated from the model environment.

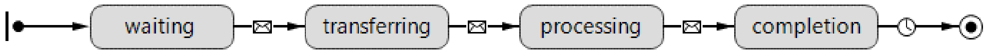


Figure 6.16 - Order's state chart

In Figure 6.17 it is possible to observe the representation of the agent in AnyLogic environment.

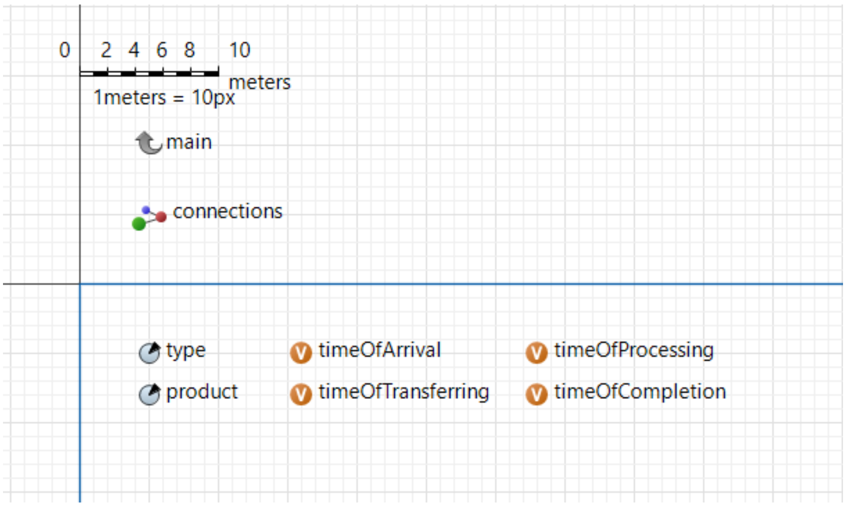


Figure 6.17 - Agent Order in AnyLogic environment

6.4 *Top-level agent*

Main is the top-level agent which representing the highest level of abstraction acts as an environment. For this reason, it embeds all the other agents living in the model and it also contains the majority of the space markup elements, the processes' flowcharts, the window for the 3D view and the statistical instruments. In order to better explain the composition and behavior of this agent, it is divided into sub-groups and each one is analyzed.

6.4.1 Space markup elements

As defined in this chapter, space markup elements are employed to characterize the environment of the model and to define agent locations. They basically reproduce the 2D layout of the warehouse. The configuration of the space markup elements in agent Main is represented in Figure 6.18.

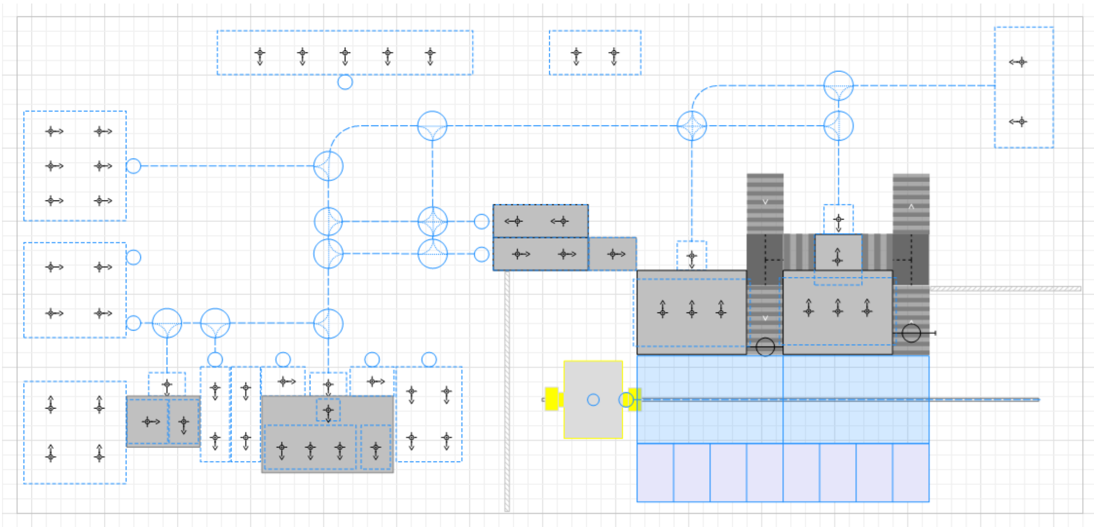


Figure 6.18 - Space markup elements

6.4.2 Elements

Main is a very complex agent and it embeds a lot of elements. Firstly, four resources blocks are introduced (see Figure 6.19). The first block describes the fleet of AGVs. This AnyLogic element allows the definition of the capacity of the fleet and the technical characteristics of the vehicles, like dimensions, speed, turning radius and minimum distance to obstacle. The second block generates a system of rack. Since the structure of the automated area, two different rack are introduced within the model, one for the side to the left of the input conveyor and one for the side to the right of the input conveyor. The two remaining blocks define the two workers of the warehouse. The operators are managed separately since they are assigned to different processes.

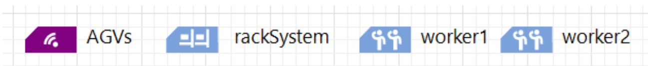


Figure 6.19 - Resources blocks

Additionally, Main is characterized by a series of parameters, variables, and collections analyzed in Table 6.11.

Table 6.11 - Description of agent Main's elements

Name	Element	Java type	Description
kitRate	Parameter	Double	Rate at which a kitting order is generated
assemblyRate	Parameter	Double	Rate at which an assembly order is generated
pickingRate	Variable	Double	Rate at which a picking order is generated
box	Variable	Box	Box currently processed at receiving station
kittingBox	Variable	Tote	Tote currently used to create a kit
product	Variable	Product	Product currently assembled at assembly station
kitCompleted	Variable	Boolean	It determines if a kitting operation is concluded
emptyChute	Variable	Boolean	It determines if the gravity plane of the kitting station is full
supplyCompleted	Variable	Boolean	It determines if the supply is completely processed
stockCompleted	Variable	Boolean	It determines if there are all the type of components at stock
stockC1	Variable	Integer	Number of C1 at stock
stockC2	Variable	Integer	Number of C2 at stock
stockC3	Variable	Integer	Number of C3 at stock
stockKitP1	Variable	Integer	Number of kits P1 at stock
stockKitP2	Variable	Integer	Number of kits P2 at stock
stockKitP3	Variable	Integer	Number of kits P3 at stock
stockP1	Variable	Integer	Number of P1 at stock
stockP2	Variable	Integer	Number of P2 at stock
stockP3	Variable	Integer	Number of P3 at stock
P1Assembled	Variable	Integer	Number of P1 assembled
P2Assembled	Variable	Integer	Number of P2 assembled
P3Assembled	Variable	Integer	Number of P3 assembled
finalProducts	Collection	ProductType	Array list containing all the type of products handled within the warehouse
emptyTotes	Collection	Tote	List of empty totes stored

Name	Element	Java type	Description
materialStored	Collection	Tote	List of totes containing components stored
materialStored1	Collection	Tote	List of totes containing components stored in rack 1
materialStored2	Collection	Tote	List of totes containing components stored in rack 2
materialRetrieved	Collection	Tote	List of totes retrieved
completedKit	Collection	ProductType	List of kits completed
kitStored	Collection	Tote	List of totes containing kit stored
queueKitting	Collection	Order	List of kitting orders in queue
transferringKitting	Collection	Order	List of kitting orders taken over
processingKitting	Collection	Order	List of kitting orders in processing
completedKitting	Collection	Order	List of kitting orders completed
queueAssembly	Collection	Order	List of assembly orders in queue
transferringAssembly	Collection	Order	List of assembly orders taken over
processingAssembly	Collection	Order	List of assembly orders in processing
completedAssembly	Collection	Order	List of assembly orders completed
queuePicking	Collection	Order	List of picking orders in queue
transferringPicking	Collection	Order	List of picking orders taken over
processingPicking	Collection	Order	List of picking orders in processing
completedPicking	Collection	Order	List of picking orders completed

Furthermore, a set of functions and events are introduced within the model. These are described in Table 6.12. The Java code of all the functions is available in Attachment I - Functions.

Table 6.12 - Description of agent Main's functions and events

Name	Element	Description
navigate	Function	It allows the navigation through the model during simulations
getFreeCell	Function	It finds a free cell within the rack and reserve it for the following storage operation
supplyOrder	Function	It generates a supply order for the type of component needed
quantityDropoff	Function	It calculates the exact quantity of components to be removed from the totes stored in the kitting station's gravity planes
quantityPickUp	Function	It calculates the quantity to be inserted into totes at the receiving station

Name	Element	Description
selectBox	Function	It selects a tote that needs a refill operation or an empty tote for the receiving process
stockControl	Function	It controls the stock levels of the components and it launches the function supplyOrder if necessary. It is called at every storage or retrieval operation
flow_Control	Function	It balances the material flows within the model environment in order to avoid system blocks
checkStoredKit	Function	It checks the presence of a kit stored within the rack of the typology requested by the assembly order. If it is present, a retrieval task is sent to mini-load, while it is not present a kitting order is created
selectStoredKit	Function	It selects the stored kit to be retrieved by mini-load, obtaining the exact coordinates of it
kittingOrder	Function	It sends a retrieval task for an empty tote and it controls the presence of enough totes stored in the gravity planes of the kitting station. If there are not sufficient totes, as many retrieval tasks as necessary are sent to mini-load
selectEmptyBox	Function	It selects the empty tote to be retrieved by mini-load, obtaining the exact coordinates of it
selectKittingBox	Function	It selects the tote containing components to be retrieved by mini-load, obtaining the exact coordinates of it
checkStoredProduct	Function	It checks the presence of a tote containing products stored within the rack of the typology requested by the picking order. If it is present, a retrieval task is sent to mini-load, while it is not present an assembly order is created
selectPickingBox	Function	It selects the tote containing products to be retrieved by mini-load, obtaining the exact coordinates of it
selectAssemblyBox	Function	At assembly station, it selects in which tote has to be inserted the new assembled product
assignAssemblyBox	Function	At assembly station, it assigns the typology of product to a tote when it is replaced with an empty one
kittingEvent	Event	It generates a kitting order, adding it to the queue of kitting orders. The logic of order generation establishes that the kit to be created is of the typology of products in minor quantity at stock. It occurs at a rate equivalent to the value of the parameter kitRate

Name	Element	Description
assemblyEvent	Event	It generates an assembly order, adding it to the queue of assembly orders. The logic of order generation establishes that the product to be assembled is of the typology of products in minor quantity at stock. It occurs at a rate equivalent to the value of the parameter assemblyRate
pickingEvent	Event	It generates a picking order, adding it to the queue of picking orders. The logic of order generation is random. It occurs at a rate equivalent to the value of the parameter pickRate
flowControl	Event	It launches every 0.5 seconds the function flow_Control

6.4.3 Model initializations

At model start-up, there is the necessity to make some initializations. One of them is the creation of a first supply of components since at the beginning of the simulation the warehouse is empty. The flow chart that represents a supply of components is showed in Figure 6.20. It is constituted by three identical sub-blocks, that is two sources, two queues, one hold and a pick-up. The first source creates the boxes that will contain the components, while the second source generates the components itself. After been created, the value of parameter type in relation from which source they exit is assigned to the components (see Table 5.1). They are subsequently inserted into the boxes thanks to the pick-up block. The three sub-blocks are all connected to the same exit port, which sends the agents to the receiving flow chart (Figure 6.24). At model start-up, the function supplyOrder is called three times, one for each type of component (see Attachment I - Functions).

This flow chart is also used during simulation in case of another supply of components is needed.

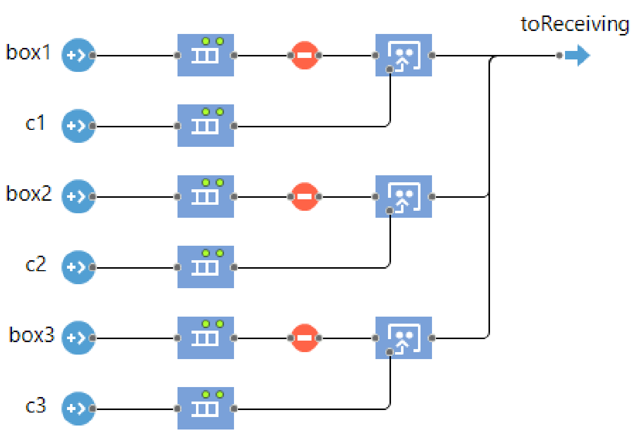


Figure 6.20 - Supply of components flow chart

Additionally, at model initialization the rack is charged with some totes containing products. This aspect is introduced within the model in order to let the picking process to start properly once the initial supply is completed. The flow chart of products' initialization is represented in Figure 6.21. Its structure is basically identical to the supply flow chart, since the basic principles are the same. Three are the only differences. Firstly, the sources that generated boxes are substituted with sources that generate totes. Secondly, the sources that generated components are replaced with sources that generate products. Thirdly, the exit port does not send the agent to the receiving station, but it sends the agents to enter3. After entering enter3, the agents are stored within the rack mean the dedicated block.

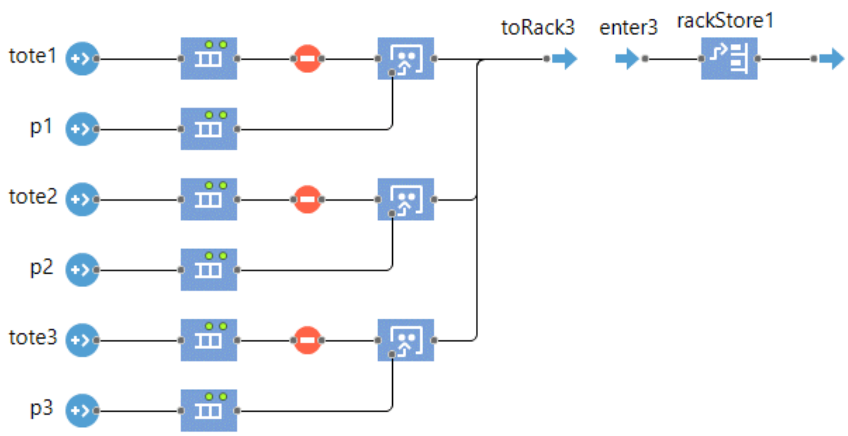


Figure 6.21 - Initialization of products

Another initialization consists of the storage of empty totes within the rack. It is made by the flow chart represented in Figure 6.22. Thanks to Java code, at model start-up 6 new totes are sent through enter5 and stored at dedicated positions in the rack system.

Figure 6.22 - Initialization of empty totes

In Figure 6.23 it is possible to observe the flow chart dedicated to the general buffer of empty totes. At model start up, 60 new empty totes are sent through enter7, filling the buffer represented by the wait block bufferEmpty. It is followed by a hold block that is used to manage the material flow through the warehouse. In fact, if there is not enough space to store another tote within the rack, the flow_Control function sets it to blocked (see Attachment I - Functions). Additionally, the block moveToRack is needed to transfer the totes to the rack by AGV. Every time that an empty tote is retrieved from the rack, another one is released from bufferEmpty and sent to rack.

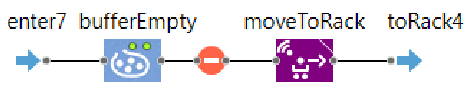


Figure 6.23 - Buffer of empty totes flow chart

Finally, at model start-up the other buffers present in the warehouse are charged with totes. These are the buffer of empty totes at the receiving station (20 totes), the buffer of empty totes at the picking station (2 totes), the buffer of empty totes at assembly station (4 totes), and the buffer of totes that will contain the assembled products at assembly station (3 totes).

6.4.4 Receiving process

The receiving process is represented by the flow chart in Figure 6.24. The source generates a specific number of empty totes at model initialization (20 totes). The port enter receives the empty totes from other processes, while enter1 receives the supply boxes containing the components. These boxes enter the block queueBoxes. Only one box at the time can enter the delay1 block. When the agent Box enters the delay block, the function selectBox is called (see Attachment I - Functions). It is employed to select a tote from the buffer. If there is a tote that needs refill, this one is released from waitTote, while if there is not a tote that needs refill, an empty tote is selected. The agent Tote flows into the block seizeWorker. The assigned worker for this operation (worker 2) moves towards the receiving station. Once the operator reaches the station, the agent Box exits delay1 and it is moved on the working table. Subsequentially, also the tote is moved to the working table. At this point, the agent Box flows into dropFromBox, which permits to remove the components from the box. The components enter the block bufferComponets and they are inserted into the selected tote through pickupInTote. Here, the function quantityPickUp is launched in order to determine the quantity to insert into the tote (see Attachment I - Functions).

Then, the agent Tote is delayed for the time assigned to this process, it is moved to the buffer and the resource is released. At this point, if the supply box is empty it is removed from the model space, while if it still contains some components another tote is selected. If there is enough space within the rack, the last hold block turns into unblocked and the totes are moved to the buffer. Then, they are transported by AGV to the automated area thanks to moveToRack1.

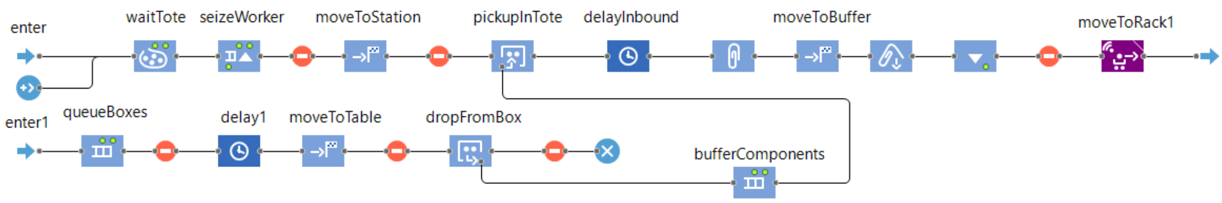


Figure 6.24 - Receiving process flow chart

6.4.5 Automated area

The functioning of the automated area is represented in Figure 6.25. Three are the possible enters to the automated area. The first is from the picking station, the second is from the AGVs, and the third is from the kitting station. All the three enter blocks are followed by a convey block that moves the tote from starting point to the interface point with the mini-load stacker crane. Once the tote arrives at the end of the input conveyor, a message is sent to the mini-load stacker crane. This message can be a storage or a transfer task. In case of a storage operation, the function getFreeCell is called in order to find a free cell to store the tote (see Attachment I - Functions).

When the mini-load stacker crane arrives at input conveyor, holdLeave tuns into unblock, the tote exits it and it is moved to the mini-load stacker crane through moveToCrane block. Then, the agent exits the Main through the exit port. It returns to Main to be stored though rackStore2 block. It is used a queue block instead of the specific block since the Java method to perform the storage operation is manually written in it.

The two unnamed hold blocks are employed to manage the material flow within the warehouse through the flow_Control function (see Attachment I - Functions). In fact, if the output conveyor is full of totes, the agents coming from the kitting station and destined to assembly cannot be transferred to the output conveyor.

Finally, the time measure start and time measure end blocks are employed in the collection of data for the statistical analysis of the performance of the warehouse. In particular, they allow the calculation of the storage and transfer cycle times.

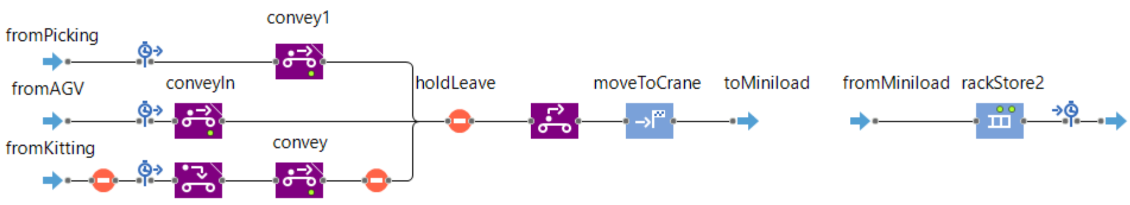


Figure 6.25 - Automated area flow chart

6.4.6 Kitting process

The kitting process flow chart begins with an enter port that takes agents from Crane. In relation to the nature of the tote retrieved, it enters in a specific branch of the flow chart. In particular, it enters out_1 if the tote is empty, while it enters in out_2 if the tote contains some components. After being entered in out_1, the tote is moved to the output conveyor through the block moveToConveyor. If the tote is destined to kitting, it is conveyed to the kitting station through conveyToStation, while on the other hand it is transported to the end of the output conveyor though agv_out. In this second case, the tote can be subsequentially transferred to the receiving station or to the assembly station by AGV. Once the empty tote arrives to the kitting station, the resource is seized (worker 2). When the operator arrives at position and if there are all the totes in the gravity planes, the kitting operation is performed. The totes containing components that are moved into the gravity planes through moveToChuteKitting wait in waitChute until this moment. When the kitting operation may start, they are all released and enter dropoff1. Here, the specific quantity of components for each tote to be picked up is calculated through the function quantityDropOff (see Attachment I - Functions).

Then, if the tote contains a sufficient quantity of components (major than 4), it returns back to waitChute, while on the other hand it is moved to the input conveyor and destined to refill or storage operations. Once the components are removed from the tote, they are inserted into an empty tote though kittingProcess block. After that, the resource is released, and the tote enters the input conveyor. In relation to the assigned process, it is then transferred out or stored within the rack.

At completion of the kitting operation, it is checked the situation of the queues of assembly and kitting orders. If there are assembly orders waiting the function checkStoredKit is called, while if there are kitting orders waiting the function kittingOrder(kitToStore) is launched (see Attachment I - Functions).

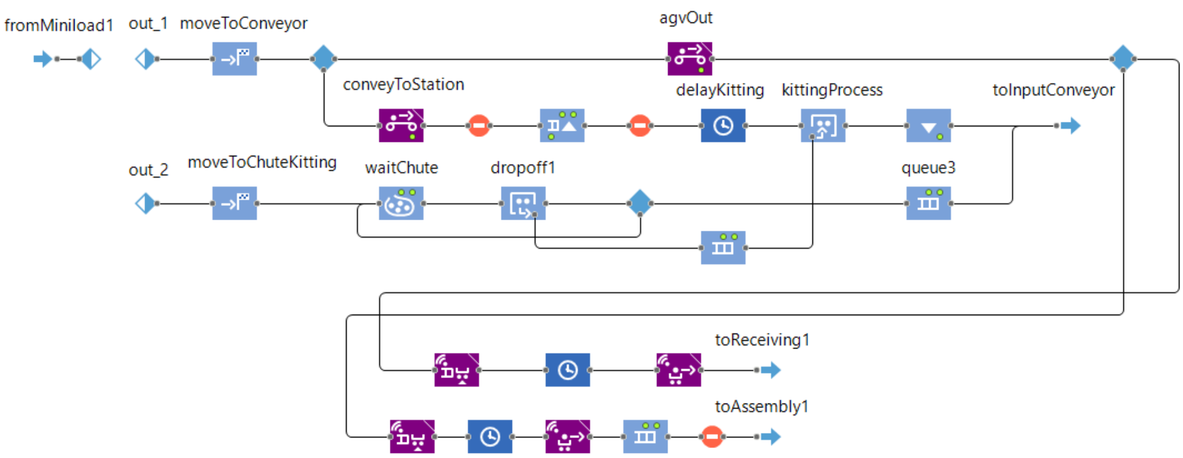


Figure 6.26 - Kitting process flow chart

6.4.7 Assembly process

The assembly process is represented in Figure 6.27. The tote containing the kit to be assembled and coming from the automated area enters the chart at the port fromRack. The assigned resource (worker 1) is seized and starts moving in direction of the assembly station. Once at position, the assemble process can start. The kit contained in the tote is removed from it and it is assembled in the product through the block assemblyProcess. Once terminated the time assigned to the operation, the assembled product enters queue10 and then it is inserted into a specific tote though pickup6. The tote in which put the product is selected by the function selectAssemblyBox and released from toteTable (see Attachment I - Functions).

After this step, if the tote contains a sufficient number of products it is sent to the rack (equal to 10), while on the other hand it returns to the block toteTable. In case the tote is moved to the automated area, there is the need to replace it. For this reason, a tote is released from emptyAssembly and the function assignAssemblyBox is called (see Attachment I - Functions).

The tote containing the kit is now empty and it can be sent to different destination in relation to the capacity conditions of the buffers present in the warehouse. It can be put in buffer near the assembly station if it flows through o1, it can be transported by the operator to the picking station if it enters o2, it can be transferred by AGV to the rack or to the general buffer if it goes through o3, and it can be moved by the operator to receiving station if it flows though o4.

Concluded the assembly operation, if there are other assembly orders waiting, the function checkStoredKit is called (see Attachment I - Functions).

The two sources generate empty totes destined to the buffer (4 totes) and to the table (3 totes) at model initialization.

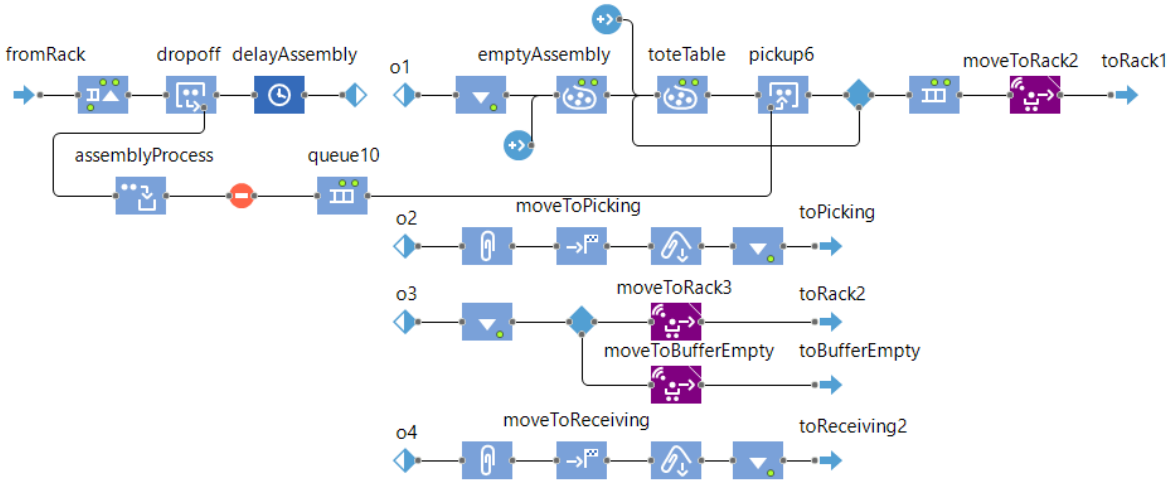


Figure 6.27 - Assembly process flow chart

6.4.8 Picking process

In Figure 6.28 it is depicted the picking process. At model initialization, the source generates two empty totes to fulfil the buffer present at the station. Out_3 receives the agents from Crane, which are moved to the gravity planes at the station thanks to moveToChutePicking. At arrival of a tote to be processed, a message is sent to the assigned resource (worker 2), who starts moving in direction of the station. Once at position, the picking process is performed. An empty tote exits the block bufferPicking and it is moved to the table near the station. The number of products requested in the order is picked up from the tote though dropoff2. They are subsequently inserted in the empty tote through the block pickingProcess. After the completion of the entire operation, the tote in the gravity plane is inserted in the input conveyor if it still contains some product, while on the other hand it is moved to the buffer of empty totes of the station. The tote containing the processed orders is transferred to the output buffer, represented by the block bufferPickingOut. If the buffer is full, that is it contains two totes, the resource moves these two totes to the outbound area. When it contains 6 elements, it is emptied, and the totes exit the model environment simulating a shipping process.

At picking process completion and if there are some picking orders waiting, the function checkStoredProduct is called (see Attachment I - Functions).

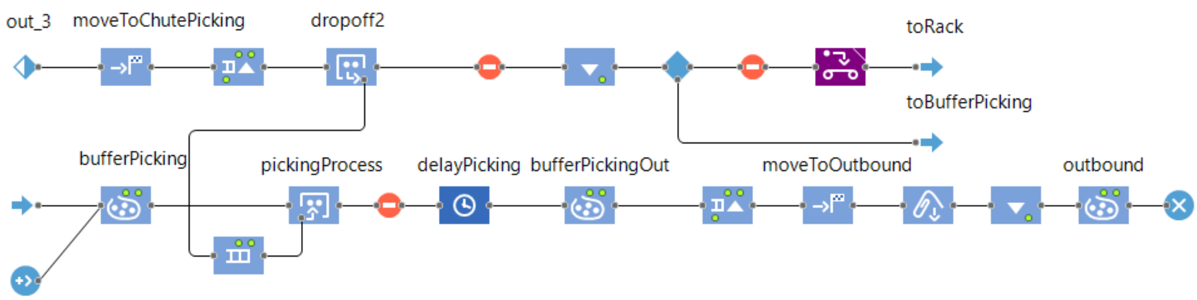


Figure 6.28 - Picking process

6.4.9 3D window

The 3D window is an embedded element of the software that allows to transform the 2D space markup elements in 3D objects during simulation. It permits to better analyze and understand the behavior of the system while performing all the processes. The 3D view offered by the 3D window during simulation is observable in Figure 6.29.

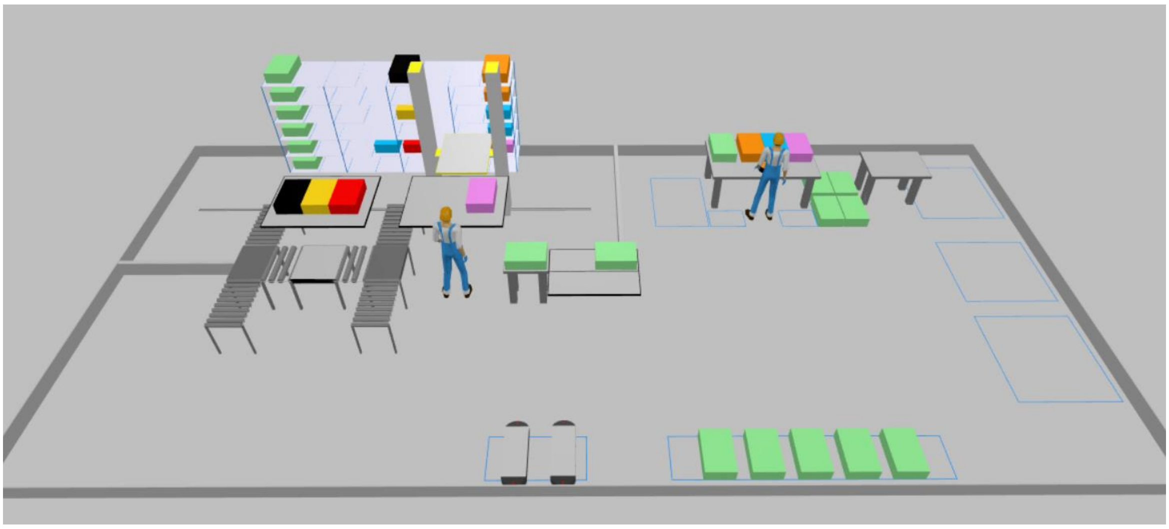


Figure 6.29 - 3D view of the model

6.4.10 Statistics

AnyLogic offers some tools to collect statistics and analyze them, such as data sets and graphs. Datasets allow the storage of data, while graphs permit to represent graphically some statistics of the datasets. Additionally, AnyLogic possesses some Java methods that consent to directly calculate the desired statistics. This is the case of utilization statistics, calculable with the method `utilization()` in case of resources, and `statsUtilization.mean()` in case of delays. The utilization of the rack is manually calculated considering the occupied space in relation to the total capacity.

The mini-load statistics are collected though time measure start and time measure end blocks. These two blocks automatically generate a dataset. To calculate and represent the mean value of the relative dataset, it is only necessary to call the Java method `dataset.getYMean().`

For what concern kitting order statistics, three datasets are inserted in the model:

- kittingOrderWaiting
- kittingOrderTransferring
- kittingOrderProcessing

KittingOrderWaiting stores the waiting times of kitting orders, kittingOrderTransferring collects the transferring times of kitting orders, and kittingOrderProcessing accumulates the processing times of kitting orders. Every second it is calculated the mean value of these datasets though the Java method `getYMean().` Furthermore, kitting cycle time is calculated every second adding the mean value of the kittingOrderWaiting and kittingOrderTransferrring datasets, while kitting order time is calculated every second adding every second the mean value of kittingOrderWaiting, kittingOrderTransferrring and kittingOrderProcessing datasets. The same procedure is followed for assembly and picking orders.

In order to collect the remaining statistics, that is WIP and throughput per hour, a function called statistics is launched every 1 second (see Attachment I - Functions). This function allows to calculate WIP and throughput per hour of the mini-load stacker crane, kitting, assembly and picking and charge the values in dedicated datasets.

The mean values of the data sets and the other statistics can be represented in bar charts, in order to better visualize them during simulation, as it is possible to observe in Figure 6.30.

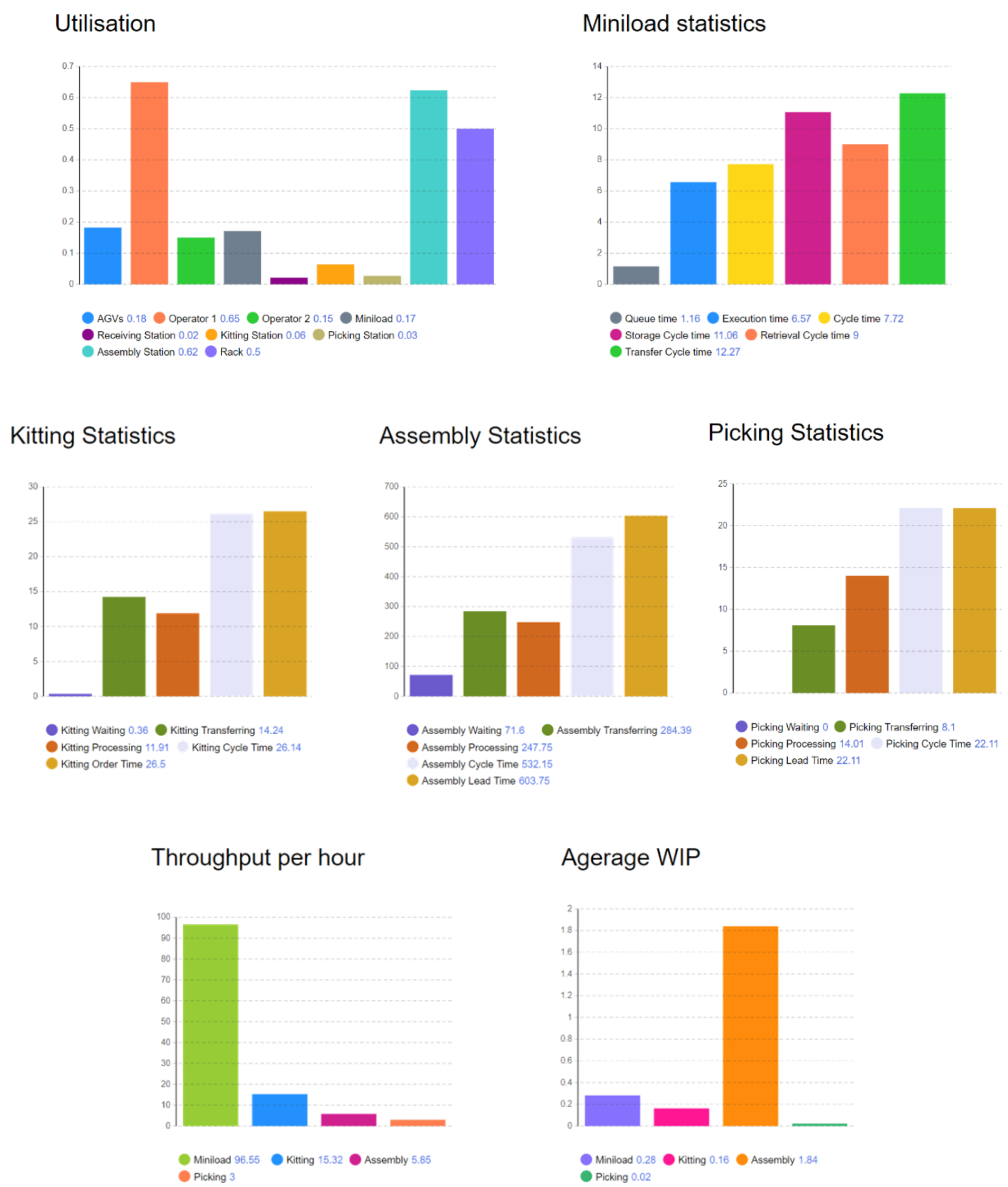


Figure 6.30 - Statistics dashboard

7. Experiments and results

This chapter aims to describe the experiments conducted on the model and the results obtained by the various simulations. The limited version of AnyLogic allows only simulations that last at most 1 hour (model time) and this may generate outcomes that are not too statistically correct. For this reason, to partially mitigate the uncertainty of results, 5 different simulations are conducted for each experiment, the values of KPIs are collected and then an average value is calculated. In addition, it is necessary to take into consideration the possibility that the warehouse may not reach the state of operational regime in the simulation time allowed by the software.

In this chapter are presented only the final aggregated results of the various simulations. For further detail see Attachment II – Experiments results.

7.1 Experiment 1: Base model

The first experiment conducted refers to the model as it is described in chapter 6 - The virtual model. The input values of the demand rates are set as follow:

- Kitting: 13 operations/hour;
- Assembly: 12 operations/hour;
- Picking: 8 operations/hour;

The results of this experiment are shown in Table 7.1. It is possible to observe that there is a consistent difference in utilization between the two human resources, that also reflects in a substantial disparity in utilization of kitting station (assigned to worker 2), picking station (assigned to worker 2) and assembly station (assigned to worker 1).

For what concern the mini-load stacker crane there are no alarming values about cycle times. The minimum difference between storage, retrieval, transfer cycle time is given by the distance travelled by the stacker crane and the mean time that the operation waits to be performed.

Regarding the three main processes of the warehouse, that is kitting, assembly and picking, it is possible to notice that the major criticalities reside in assembly. This is due to different reasons. Firstly, the assembly process is the one that present the longest execution times. Furthermore, the assembly station is located quite far from the automated area causing an increase in transferring times.

The work in progress values show that the mini-load stacker crane, the kitting station and the picking station do not always have an order to perform, while the assembly station has on average more than one tote to process at the same time.

The mini-load stacker crane is the element within the warehouse that presents the highest throughput in terms of operations per working hour, thanks to short cycles time and high number of operations. On the other hand, the kitting, assembly and picking throughputs have lower values due to the characteristics of the processes. They do not correspond exactly to the input values of simulations (rates) since the impossibility for the model to reach a total operational regime state. Additionally, the kitting throughput is influenced by the kitting order that are automatically generated when there is not a stored kit within the rack. The same event occurs with the assembly throughput, which is affected by the assembly orders that are produced when a picking order could not be fulfilled due to the absence of the relative products stored within the rack.

Table 7.1 - Results experiment 1

Category	KPI	Value
Utilization [%]	AGVs	14.4%
	Worker 1	65.8%
	Worker 2	13.8%
	Mini-load	14.6%
	Receiving station	13.0%
	Kitting station	5.4%
	Picking station	2.0%
	Assembly station	64.6%
	Rack	44.2%
Mini-load [s]	Queue time	1.43
	Execution time	6.57
	Cycle time	8.00
	Storage lead time	10.96
	Retrieval lead time	9.38
	Transfer lead time	12.70
Kitting [s]	Kitting waiting	5.57
	Kitting transferring	13.14
	Kitting processing	11.64
	Kitting cycle time	24.78
	Kitting order time	30.35

Category	KPI	Value
Assembly [s]	Assembly waiting	46.48
	Assembly transferring	173.93
	Assembly processing	232.65
	Assembly cycle time	406.58
	Assembly order time	453.06
Picking [s]	Picking waiting	0.11
	Picking transferring	12.85
	Picking processing	13.94
	Picking cycle time	26.79
	Picking order time	26.90
Average WIP [n° of items]	Mini-load	0.18
	Kitting	0.14
	Assembly	1.45
	Picking	0.05
Throughput per hour [ops/hour]	Mini-load	74.33
	Kitting	11.70
	Assembly	6.41
	Picking	3.57

7.2 Experiment 2: Modified rates

Starting from the analysis of the results of the first experiment and trying to improve them, a second experiment is conducted with modified rates. Since the picking and the kitting stations present low values of utilization in the first experiment, the picking and kitting rates are increased. It is expected that increasing the kitting rate the number of kits stored within the rack will enhance. This may accelerate the transferring process to the assembly station since the probability to find an already created kit in higher and there will be not the necessity to create a kit on request. The input values of the model are set as follow:

- Kitting: 20 operations/hour;
- Assembly: 12 operations/hour;
- Picking: 20 operations/hour;

Table 7.2 summarizes the results of the various simulations.

It is possible to notice variations in the values of KPIs in relation to experiment 1. A general increase in the values of utilization occurs due to an increased amount of orders to process. The augmented picking rate leads to an enhanced utilization of both picking and assembly stations. This is expected since the higher number of picking orders determines the necessity to a higher stock of products generated by the assembly process. This also causes a great amplification in the assembly order time due to a too high workload for the assembly station.

The kitting order time increases due to a greater time spent by kitting orders waiting to be processed. The cause of that resides in the fact that the worker assigned to the kitting process is also in charge of the picking one. Consequently, a higher picking rate leads to major occupancy of the operator in picking operations with subsequent increase in the waiting time of kitting orders. The same event is observable in the augmented picking order time.

Clearly, the increased workload also determines an enhancement in WIP and throughput values.

The mini-load stacker crane statistics remain stable.

Observing the trend of the KPIs during this experiment, it seems to be that both the picking and assembly order times continue to increase. This is caused by the impossibility to dispose of the picking orders due to the absence of enough products stored within the rack. Because of that, a sequence of assembly orders is generated. These can be not processed by the assembly station due to high values of processing time. This cause a continuous increase of the number of assembly orders in queue that may grow to infinity values on a long-term basis.

Table 7.2 - Results experiment 2

Category	KPI	Value
Utilization [%]	AGVs	20.6%
	Worker 1	78.0%
	Worker 2	25.2%
	Mini-load	22.8%
	Receiving station	2.0%
	Kitting station	7.6%
	Picking station	7.6%
	Assembly station	74.8%
	Rack	48.0%
Mini-load [s]	Queue time	1.55
	Execution time	6.56
	Cycle time	8.11
	Storage lead time	11.04
	Retrieval lead time	9.32
	Transfer lead time	12.68
Kitting [s]	Kitting waiting	33.63
	Kitting transferring	14.06
	Kitting processing	11.72
	Kitting cycle time	25.78
	Kitting order time	59.41

Category	KPI	Value
Assembly [s]	Assembly waiting	50.89
	Assembly transferring	333.04
	Assembly processing	234.06
	Assembly cycle time	567.10
	Assembly order time	617.99
Picking [s]	Picking waiting	6.59
	Picking transferring	12.46
	Picking processing	14.00
	Picking cycle time	26.46
	Picking order time	33.05
Average WIP [n° of items]	Mini-load	0.28
	Kitting	0.40
	Assembly	2.93
	Picking	1.16
Throughput per hour [ops/hour]	Mini-load	119.69
	Kitting	18.18
	Assembly	7.40
	Picking	13.72

7.3 Experiment 3: Modified layout

In order to overcome the problems faced during experiment 2, a third experiment is proposed. It consists of the modification of the layout of the warehouse for what concerns the zones external the automated area.

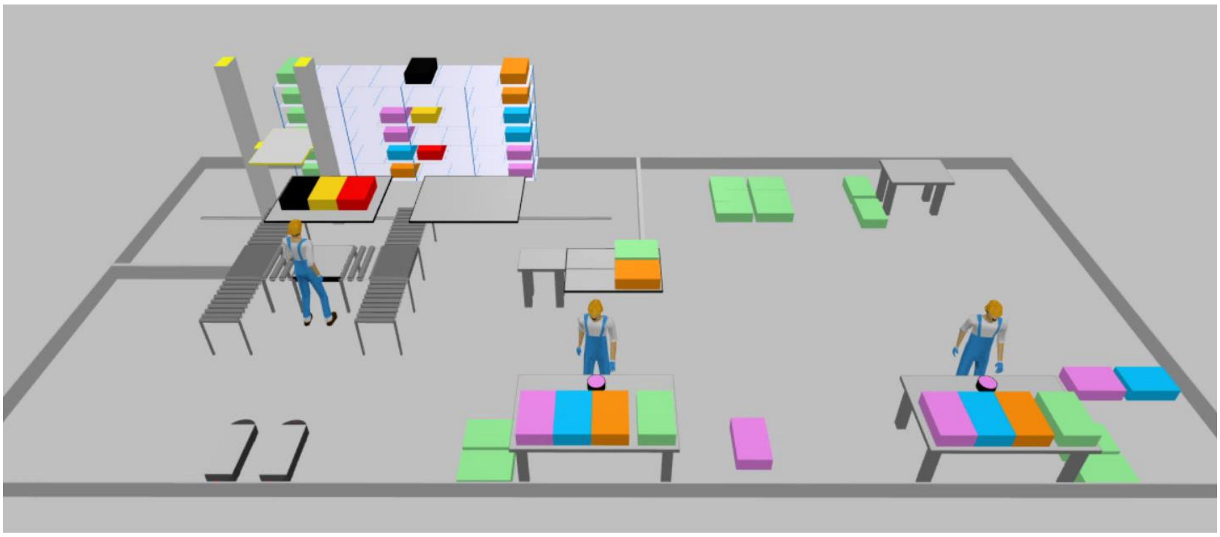


Figure 7.1 - Modified layout

As shown in Figure 7.1, another assembly station is added to the environment and the capacity of the buffer of totes coming from the automated area containing the kit to be assembled is doubled, in order to ensure greater volume of work for both stations. Furthermore, the presence of another operator is assumed. The position of the assembly station is switched with the one of the buffer of empty totes, which is moved near the receiving station. The AGVs' home is slightly moved in front of the automated station.

To understand if the only changes in layout may solve the problems of experiment 2, the input values of the model are maintained constant, that is:

- Kitting: 20 operations/hour;
- Assembly: 12 operations/hour;
- Picking: 20 operations/hour;

Utilization statistics for the added worker and assembly station are included within the model. The results of various simulations are summarized in Table 7.3.

Confronting the results between experiment 2 and experiment 3, it is possible to observe a general improvement of the KPI values. The AGVs utilization does not change significantly, while a decrease in utilization of worker 2 occurs. This is due the split of assembly orders between worker 2 and worker 3, which reaches a utilization of 41%. Worker 2 slightly increase its percentage of time occupied, but only by a very low margin. Also mini-load stacker crane's utilization grows a little, while receiving, kitting and picking stations remain occupied more or less the same amount of time. There is an expected decrease in utilization of the first assembly station, thanks to the introduction of the second assembly station. Also the rack shows a slightly enhancement of utilization value.

Table 7.3 - Results experiment 3

Category	KPI	Value	Category	KPI	Value
Utilization [%]	AGVs	22.4%	Assembly [s]	Assembly waiting	20.09
	Worker 1	57.4%		Assembly transferring	92.07
	Worker 2	28.2%		Assembly processing	245.01
	Worker 3	41.4%		Assembly cycle time	337.08
	Mini-load	27.0%		Assembly order time	357.17
	Receiving station	2.2%			
	Kitting station	9.2%	Picking [s]	Picking waiting	34.46
	Picking station	7.6%		Picking transferring	13.58
	Assembly station 1	51.6%		Picking processing	14.06
	Assembly station 2	38.0%		Picking cycle time	27.66
	Rack	54.2%		Picking order time	62.32
Mini-load [s]	Queue time	2.05			
	Execution time	6.56	Average WIP [n° of items]	Mini-load	0.39
	Cycle time	8.62		Kitting	0.40
	Storage lead time	12.25		Assembly	1.54
	Retrieval lead time	9.56		Picking	1.23
	Transfer lead time	13.32	Throughput per hour [ops/hour]	Mini-load	144.31
Kitting [s]	Kitting waiting	24.40		Kitting	23.50
	Kitting transferring	15.80		Assembly	7.75
	Kitting processing	11.64		Picking	15.18
	Kitting cycle time	27.44			
	Kitting order time	51.84			

For what concerns the mini-load stacker crane, the statistics remain more or less the same, showing only a small increase in queue time and lead time of all the operations. These changes are not so significant.

Observing the kitting station, it is possible to state that the overall situation is improved, since the kitting order time value reduces. A significant upgrade occurs in assembly process, where the assembly order time decreases by 42%, thanks to an important contraction in assembly waiting and transferring times. This phenomenon is generated by a greater availability of the two assembly stations that causes a reduction of waiting time for an order before it is taken over. Furthermore, the possibility to split the assembly workload between two station lowers the transferring time.

On the other hand, it is possible to observe a deterioration in picking statistics in terms of waiting time. It is possible that this is caused by the greater number of supplies and kitting orders that worker 2 has to handle. The increased amount of supplies is generated by a bigger consumption of materials for the creation of kits destined to assembly. Furthermore, the worsening of the picking order and cycle times may be due the split of assembly orders between the two stations, which causes an augmented time before the completion of a tote at assembly stations and its transfer to the rack.

Regarding the WIP, the assembly process shows a reduction of the KPI around 48%, mini-load stacker crane and picking process are characterized by slightly increase, while kitting process remains constant.

For what concerns throughput, all the voices show an increase in values. The smallest improvement is present in assembly operations. According to the utilization and order time values, this suggests the possibility to increase the assembly rate in input.

7.4 Experiment 4: Modified priorities

Starting from the results obtained in experiment 3, a fourth experiment is proposed. It basically consists of the modification of some priorities within the model. Since in experiment 3 the picking process suffers some deterioration in KPIs, in experiment 4 the priority of the process is increased, forcing the operator to work on a picking order even if it has arrived after other orders. It is expected that this will cause an increase in kitting order time, due to a growth in waiting time.

In addition, it may happen that an intensification of the output of products generates a quicker decrease in stock levels to the point that the picking orders themselves are not processable. For this reason, it is introduced a simple scheduling of the picking orders through a function called schedulePickingOrder (see Attachment I - Functions). In case a picking order is not processable due to a lack of products in stock, the system verifies the possibility to complete another subsequent order with the products stored. If the control is positive, the correspondent order is shifted on the top of the queue and all the other orders are moved by one position to the bottom of the queue. This operation could generate an improvement of some percentage point on the waiting times of the picking orders.

Finally, since the statistics regarding assembly orders and assembly stations are positive and present a margin of improvement, the assembly rate is increased in order to increment the input of manufactured product in direction of the rack.

So, the input values of the model are:

- Kitting: 20 operations/hour;
- Assembly: 20 operations/hour;
- Picking: 20 operations/hour;

The results of the various simulations are summarized in Table 7.4.

Starting from the analysis of utilization values, it is possible to notice that the principal variations in comparison with experiment 3 regard worker 1 and worker 3, the two assembly stations and the rack. These are expected values since the increment of the assembly rate causes a growth in utilization of the resources allocated on assembly operations, as well as of the assembly stations. The diminution of rack utilization is generated by a greater retrieval of kits and products destined to assembly and picking stations respectively, thanks to the introduced scheduling. To confirm that, it is observable a slightly improvement in the statistics of the picking station. In contraposition to that, a small decrease in utilization of kitting station occurs.

For what concerns mini-load statistics, no significant changes in KPIs values are noticeable.

Regarding the kitting orders, a small diminution of the order time is observable. It is caused by a decrease of the waiting time, probably generated by the greater availability of the station. Despite that, the change in statistics is not so relevant.

As expected, the assembly order time worsens. The waiting time slightly reduces, the transferring time grows by 113%, while the other statistics remain stable. This generates an increment of cycle time by 34% and of order time by 30%.

Picking statistics marginally improve thanks to a diminution of waiting and transferring times.

For what concern the WIP, the biggest changes are observable in assembly, since the KPI increases by around 100%. This result reflects the worsening in transferring time for the assembly orders, that need to wait more in the dedicated buffer before they are effectively processed by the operator.

Finally, the operations per hour performed by the mini-load stacker crane and the kitting orders processed marginally reduce, while the number of assembly and picking orders completely processed per working hour increases. The growth is generated by a greater number of assembly orders generated and the scheduling introduced.

Table 7.4 - Results experiment 4

Category	KPI	Value	Category	KPI	Value
Utilization [%]	AGVs	25.2%	Assembly [s]	Assembly waiting	14.31
	Worker 1	76.2%		Assembly transferring	207.99
	Worker 2	31.2%		Assembly processing	243.41
	Worker 3	60.4%		Assembly cycle time	451.39
	Mini-load	28.2%		Assembly order time	465.70
	Receiving station	2.0%			
	Kitting station	7.6%	Picking [s]	Picking waiting	31.88
	Picking station	8.4%		Picking transferring	11.98
	Assembly station 1	69.6%		Picking processing	14.27
	Assembly station 2	61.8%		Picking cycle time	26.24
	Rack	38.4%		Picking order time	58.12
Mini-load [s]	Queue time	1.88			
	Execution time	6.58	Average WIP [n° of items]	Mini-load	0.38
	Cycle time	8.46		Kitting	0.38
	Storage lead time	11.53		Assembly	3.09
	Retrieval lead time	9.25		Picking	1.28
	Transfer lead time	14.60	Throughput per hour [ops/hour]	Mini-load	141.48
Kitting [s]	Kitting waiting	19.19		Kitting	19.37
	Kitting transferring	15.60		Assembly	10.12
	Kitting processing	11.59		Picking	22.07
	Kitting cycle time	27.18			
	Kitting order time	46.37			

7.5 Comparison of the experiments

In Table 7.5 is shown a comparison of the results of all the experiments.

As it is possible to observe, the AGVs utilization increases in each experiment but without reaching critical values. In no experiment the automated vehicles represent a bottleneck of the system. Worker 1 shows quite high values of utilization in all the experiments. In experiment 3, the lowest value is reached thanks to the introduction of a second assembly station. The increase between experiment 3 and 4 by 18.8% is caused by the increment of the assembly order rate. Worker 2 shows a growth in utilization by 17.4% along the experiments but until not critical values. The resource is mostly of the time in idle. Worker 3 shows acceptable values of utilization. The increase between experiment 3 and 4 by 19% is caused by the increment of the assembly order rate. With the modification between the tests, the mini-load stacker crane enhances its utilization by 13.6%, but without reaching a critical value. Receiving station stays at very low utilization levels in all the experiment. Also kitting and picking station present not so high values. This is due to the impossibility to process a great amount of kitting and picking orders without overcharging the assembly stations. The same pace of worker 1 and 3 reflects in the values of assembly station 1 and 2. For what concern the rack, the highest utilization is reached in experiment 3, while the lowest in experiment 4. This last value is due to the higher turnover of materials caused by all the adjustments made during simulations.

The statistics of mini-load stacker crane do not show significant variations in the four experiments.

Not the same can be said about the kitting orders. The waiting time starts at minimum value in experiment 1, it reaches its maximum value during experiment 2 while it decreases in experiment 3 and 4. Transferring and processing times do not change significantly. Considering also the other statistics obtained during the tests, the best trade-off is reached in experiment 4.

Since the assembly process represents the most critical aspect of all the system, its statistics show great variations caused by the attempts to reach an optimal design solution. With the increment of rates between experiment 1 and 2, the cycle and orders times worse quite badly, principally due to the augmentation of transferring time. Indeed, the increase in values amounts to 39.48% and 36.40% respectively. The introduction of a second assembly station generates the best values of these statistics. The raise of the rates in experiment 4 causes an important upgrade in WIP and throughput per hour, but a deterioration of the other KPIs.

For what concern picking orders, the best values of picking cycle and order times are obtained in experiment 1. This is due to almost absent waiting times. The increment in rates causes a little increase in statistics. Waiting times become significant in experiment 3 and 4. It is possible that this is caused by the greater number of supplies and kitting orders that worker 2 has to handle. Furthermore, the split of assembly orders between the two stations may causes an augmented time before the completion of a tote at assembly stations and its transfer to the rack. Experiment 4 shows a slightly upgrade in picking statistics. Indeed, picking cycle time reduces by 5.38% and order time by 7.22%.

Regarding the WIP, the statistics improves during the experiments. The only particular result consists of the fall of the value of WIP of assembly process during experiment 3 by 90.51%. This is due to the introduction of the second assembly station without the modification of the rates. With the growth in rates of experiment 4, the WIP returns to acceptable values (+101.04%).

Finally, throughput per hour shows a general improvement during experiments. The only diminution is observable in mini-load stacker crane and kitting operations between experiment 3 and 4. Indeed, the decreases amount to 2% and 21.29% respectively. The more significant reduction in kitting throughput per hour is probably due to the picking orders scheduling introduced in experiment 4.

Table 7.5 - Comparison of results

Category	KPI	Exp. 1	Exp. 2	Exp. 3	Exp. 4
Utilization [%]	AGVs	14.4%	20.6%	22.4%	25.2%
	Worker 1	65.8%	78.0%	57.4%	76.2%
	Worker 2	13.8%	25.2%	28.2%	31.2%
	Worker 3	-	-	41.4%	60.4%
	Mini-load	14.6%	22.8%	27.0%	28.2%
	Receiving station	13.0%	2.0%	2.2%	2.0%
	Kitting station	5.4%	7.6%	9.2%	7.6%
	Picking station	2.0%	7.6%	7.6%	8.4%
	Assembly station 1	64.6%	74.8%	51.6%	69.6%
	Assembly station 2	-	-	38.0%	61.8%
	Rack	44.2%	48.0%	54.2%	38.4%
Mini-load stacker crane [s]	Queue time	1.43	1.55	2.05	1.88
	Execution time	6.57	6.56	6.56	6.58
	Cycle time	8.00	8.11	8.62	8.46
	Storage lead time	10.96	11.04	12.25	11.53
	Retrieval lead time	9.38	9.32	9.56	9.25
	Transfer lead time	12.70	12.68	13.32	14.60
Kitting [s]	Kitting waiting	5.57	33.63	24.40	19.19
	Kitting transferring	13.14	14.06	15.80	15.60
	Kitting processing	11.64	11.72	11.64	11.59
	Kitting cycle time	24.78	25.78	27.44	27.18
	Kitting order time	30.35	59.41	51.84	46.37
Assembly [s]	Assembly waiting	46.48	50.89	20.09	14.31
	Assembly transferring	173.93	333.04	92.07	207.99
	Assembly processing	232.65	234.06	245.01	243.41
	Assembly cycle time	406.58	567.10	337.08	451.39
	Assembly order time	453.06	617.99	357.17	465.70
Picking [s]	Picking waiting	0.11	6.59	34.46	31.88
	Picking transferring	12.85	12.46	13.58	11.98
	Picking processing	13.94	14.00	14.06	14.27
	Picking cycle time	26.79	26.46	27.66	26.24
	Picking order time	26.90	33.05	62.32	58.12

Category	KPI	Exp. 1	Exp. 2	Exp. 3	Exp. 4
Average WIP [n° of items]	Mini-load	0.18	0.28	0.39	0.38
	Kitting	0.14	0.40	0.40	0.38
	Assembly	1.45	2.93	1.54	3.09
	Picking	0.05	1.16	1.23	1.28
Throughput per hour [ops/hour]	Mini-load	74.33	119.69	144.31	141.48
	Kitting	11.70	18.18	23.50	19.37
	Assembly	6.41	7.40	7.75	10.12
	Picking	3.57	13.72	15.18	22.07

Conclusions

The research carried out has led to the prearranged results, that is the construction of a virtual model of the warehouse and the analysis of its performance in order to find an optimal design solution to be implemented and where reside the major criticalities of the system.

The employment of AnyLogic as environment where develop the virtual model turned out to be a good choice. It is impossible to deny the presence of initial difficulties related to the complexity of the software. As described in the chapter 6 - The virtual model, AnyLogic is a multi-method simulation software. This clearly increases the difficulty of learning its proper use. Nevertheless, once these initial difficulties were overcome, the software revealed its full potential. As can be easily deduced, the high level of complexity encountered translates into a high level of flexibility in the simulation process. The ability to integrate simulation paradigms within a single model is the strength of AnyLogic. Furthermore, the possibility to build its own agents and objects, like in the case of the mini-load stacker crane, confers added value to this software. During the construction of the virtual model, the principal eases met regard the construction of the processes through the blocks embedded in the software and the animation of them through the space markup elements. On the other hand, the principal difficulties encountered are related to the verification process, in particular for what concern the diffusion of the information. Unlike the "physical" aspect of the simulation observable through 2D and 3D windows, communication between agents, information tracking and data storage are not tangible and difficult to verify.

Regarding the result obtained from this research, the last experiment represents the best alternative tested during simulations. It is difficult to state that the one found is the optimal solution implementable for the warehouse. This is principally due to the difficulties obtained during the experiments caused by the limited version of the software. The impossibility to simulate the behavior of the system for an indefinite time causes uncertainties of the results. In the simulation time allowed by AnyLogic, it is only possible to observe the performance of the warehouse in an initial state and then to assume its behavior in a regime state on the basis of what monitored in this early phase. Furthermore, the restricted number of elements introducible within the model reduced the degree of flexibility for the construction of it. This limited the possible variations in order to test different or more articled scenarios.

Additionally, it is important to observe that some assumptions and simplifications are made during the abstraction process. There are various elements that can cause a deviation between the performance of the real system and the virtual one. Firstly, no breakdown of the machineries and maintenance are introduced within the model. Failures generally occurs in real systems and maintenance procedures are always present. Secondly, no fluctuation in demand rates is considered. Commonly in a real system, external demand is always influenced by variations, even if minimal. This clearly affects the performance of the warehouse. Thirdly, no quality processes are taken into account during the construction of the model. Quality controls influence the functioning of the system in terms of resource availability, time, rejections, and errors detection.

All the experiments show important aspects regarding the warehouse. Firstly, simulations demonstrate which are the critical points and processes. For sure, the bottleneck of the entire warehouse resides in the assembly process. In fact, it presents the worst statistics in all the tests.

Also in the best scenario found, the utilization rates of the stations and the resources are considerably greater than the other ones. This is principally due the high difference in processing time compared to the other procedures. This problem also affects negatively the other processes, which function at under-capacity levels.

Secondly, simulations illustrate which aspects of the warehouse do not represent a critical point. For sure, the receiving process is the less problematic element of the entire warehouse. The receiving station shows a very low level of utilization due to the occasional occurrence of supplies. Additionally, the high priority assigned to this process reduces to minimum delays caused by other procedures of the warehouse. Also the kitting station is not a critical point. The statistics collected during experiments are positive and present much room for deterioration before making the process problematic. The mini-load stacker crane and AGVs have low level of utilization and this suggests that they do not represent a critical aspect in the system. The margin of improvement is still high.

Thirdly, the experiments reveal which aspects of the warehouse could be potentially a problem. The picking process at initial phases of simulations does not present relevant troubles. In a state of regime of the system, picking orders' statistics may suffer considerably complications. Particularly, when the products stored within the rack terminate, picking orders may wait long times before they are processed. This is due to a low level of replenishment of products assembled. Unfortunately, this hypothesis is verifiable only with longer simulations.

In conclusion, it is possible to state that despite all the complications and problems found, the results obtained are positive and represent a good starting point for further and deeper analysis. Interesting research points certainly concern the critical aspects listed. Additionally, may be remarkable comparing the real performance of the system and the results obtained in this study after the hardware implementation of the warehouse. Finally, in order to study the optimization of the processes may be interesting create a digital twin of the warehouse. AnyLogic is a software that can be used in this type of systems. The possibility of customization enables the integration of simulation modelling, machine learning techniques and sensor data from external sources for the creation of a digital twin. It represents a powerful tool for establishing efficient operations and it allows the evaluation of the outcome of different system configurations through the analysis of a series of what-if scenarios.

Bibliography

Anu M. 1997. Introduction to modelling and simulation. *Proceedings of the 1997 Winter Simulation Conference*

Benjamin P., Erraguntla M., Delen D., Mayer R. 1998. Simulation Modeling at Multiple Levels of Abstraction. *Conference: Simulation Conference Proceedings*

Bonabeau E. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Arthur M. Sackler Colloquium of the National Academy of Sciences*

Borshchev A., Fillippov A. 2004. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. *The 22nd International Conference of the System Dynamics Society.* July 25 - 29 2004. Oxford, England

Chopra S, Meindl P. 2007. *Supply Chain Management*. 3rd edition. Stanford University, Pearson

Daum B. 2003. *Modeling Business Objects with XML Schema.* The Morgan Kaufmann Series in Software Engineering and Programming. pp. 41-70

De Vries H. J. 1999. *Standardization: A business approach to the role of national standardization organizations.* Springer Science & Business Media New York

Ellram L. M., Cooper M. C. 1990. *Supply Chain Management, Partnership, and the Shipper—Third Party Relationship.* The International Journal of Logistics Management

F. Railsback S., Grimm V. 2010. *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Princeton University Press

Fawcett S. E., Magnan G. M., Mccarter M. W. 2008. Benefits, Barriers, and Bridges to Effective Supply Chain Management. *Supply Chain Management: An international Journal.* 13/1, 35-48

Fishman G. S. 2001. *Discrete-Event Simulation: Modeling, Programming, and Analysis*. 1st ed, Springer

Fishwick P.A. 1995. *Simulation Model Design and Execution: Building Digital Worlds*. Upper Saddle River. New Jersey: Prentice-Hall, Inc.

Forrester J. 1961. *Industrial Dynamics*. Cambridge, MA: MIT Press

Gu J., Goetschalckx M., McGinnis L. F. 2009. Research on warehouse design and performance evaluation: A comprehensive review. *European Journal of Operational Research*

Gudmundsson A., Boer H., Corso M. 2004. The implementation process of standardisation. *Journal of Manufacturing Technology Management*

Håkansson, H., and Snehota, I. 1995. *Developing Relationships in Business Networks*. Routledge, London, 1995.

Houlihan J. B. 1985. *International Supply Chain Management*. International Journal of Physical Distribution and Materials Management. *MATEC Web of Conferences*

Klügl F. 2016. Using the affordance concept for model design in agent-based simulation. *Annals of Mathematics and Artificial Intelligence*

Klügl F., Bazzan A. 2012. Agent-Based Modeling and Simulation. *AI Magazine.* 33. 29-40

Kusrini E., Novendri F. Helia V. N. 2018. Determining key performance indicators for warehouse performance measurement – a case study in construction materials warehouse.

Lambert D. 2014. *Supply Chain Management: Processes, Partnerships, Performance*. Chapter 1 Supply Chain Management. Supply Chain Management Institute

Lambert D., Cooper M., Pagh J. 1998. Supply Chain Management: Implementation Issues and Research Opportunities. *The International Journal of Logistics Management*, Vol. 9, No. 2, p. 4.

Lambert D., Cooper M., Pagh J. 2000. Issues in Supply Chain Management. *Industrial Marketing Management.* 29, 65–83

Lerher T. 2016. Travel time model for double-deep shuttle-based storage and retrieval systems. *International Journal of Production Research.*

Mahdavi A. 2019. *The Art of Process-Centric Modeling.* Anylogic

MĺIkva M., Prajováa V., Yakimovichb B., Korshunovb A., Tyurinc I. 2016. Standardization - one of the tools of continuous improvement. *International Conference on Manufacturing Engineering and Materials, ICMEM 2016, 6-10 June 2016, Nový Smokovec, Slovakia*

Murray-Smith D.J. 2015. *Methods of Model Verification*. in: Testing and Validation of Computer Simulation Models. Simulation Foundations, Methods and Applications. Springer, Cham

Negri E., Fumagalli L., Macchi M. 2017. A review of the roles of Digital Twin in CPS-based production systems. *27th International Conference of Flexible Automation and Intelligent Manufacturing*

Pfohl H. C., Yahsi B., Kurnaz T. 2015. The Impact of Industry 4.0 on the Supply Chain, In: Kersten, Wolfgang Blecker, Thorsten Ringle, Christian M.: *Innovations and Strategies for Logistics and Supply Chains: Technologies, Business Models and Risk Management*. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 20, epubli GmbH, Berlin, pp. 31-58

Porter M. 1985. *Competitive Advantage: Creating and Sustaining Superior Performance.* NY: Free Press

Qi Q., Tao F. 2018. Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. *2018 IEEE*

Qinghua Z., Bo He, Guoquan C., Zhuan W., Dawei Y. and Bo W. 2010. *Development of pick-to-light communication middleware*. 2010 5th IEEE Conference on Industrial Electronics and Applications. pp. 1897-1901

Robinson S. 2008. Conceptual Modelling for Simulation Part I: Definition and Requirements. *Journal of the Operational Research Society.* pp. 278-290

Robinson S. 2011. Choosing the right model: conceptual modeling for simulation. *Proceedings of the 2011 Winter Simulation Conference*

Robinson S., Arbez G., Tolk A., Wagner G. 2015. Conceptual modelling: definition, purpose and benefits. *Proceedings of the 2015 Winter Simulation Conference*

Sargent R. G. 2011. Verification and validation of simulation models. *Proceedings of the 2011 Winter Simulation Conference*

Staudt, F. H., Di Mascolo, M., Alpan, G., Rodriguez, C. M. T. 2014. Warehouse performance measurement: classification and mathematical expressions of indicators. *ILS2014 5th International Conference*

Tao F., Zhang M. 2017. Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. *2017 IEEE.*

Thacker B. H., Deobling S. W., Hemez F. M., Anderson M. C., Pepin J. E., Rodriguez E. A. 2004. *Concepts of Model Verification and Validation*

Tjahjono B., Espugues C., Ares E., Paleaz G. 2017. What does Industry 4.0 mean to Supply Chain? *Manufacturing Engineering Society International conference, 28-30 June 2017, Vigo, Spain*

Zanjirani Farahani R., Asgari N., Davarzani H. 2009. *Supply Chain and Logistics in National, International and Governmental Environment.* Physica-Verlag, a Springer Company

Sitography

Bellinger G. 2004. Modeling & Simulation: An Introduction. *Mental Model Musings.* Available at: http://www.systems-thinking.org/index.htm

Bsaikrishna. 2016. The five major flows in supply chain. *Brandalyzer*. Available at: https://brandalyzer.blog/2016/03/23/the-five-major-flows-in-supply-chain/

Chan M. 2019. The Difference Between a B2B and B2C Supply Chain. *Unleashed*. Available at: https://www.unleashedsoftware.com/blog/difference-between-b2b-b2c-supply-chain

Chan M. 2019. Identifying the Key Differences Between a B2B and B2C Supply Chain. *Unleashed.* Available at: https://www.unleashedsoftware.com/blog/identifying-key-differences-b2b-b2c-supply-chain

Collins. Available at: https://www.collinsdictionary.com/

Council of Supply Chain Management Professionals. The importance of Supply Chain Management. Available at: https://cscmp.org/CSCMP/Develop/Starting_Your_Career/Importance_of_Supply_Chain_Management.aspx

Dictionary.com. Available at: https://www.dictionary.com/

Geissbauer R., Jesper Vedsø J., Schrauf S. 2016. A strategist's guide to Industry 4.0. *Stategy-business.* Available at: https://www.strategy-business.com/article/A-Strategists-Guide-to-Industry-4.0?gko=7c4cf

Grimshaw J. 2020. What is supply chain? A definitive guide. *Supply Chain Digital.* Available at: https://www.supplychaindigital.com/supply-chain-management/what-supply-chain-definitive-guide

Handfield R. 2020. What is Supply Chain Management (SCM)? *NC State University.* Available at: https://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management-scm

Hoey B. 2018. Transport Logistics in the Era of Industry 4.0. *Blog flexis*. Available at: https://blog.flexis.com/transport-logistics-in-the-era-of-industry-4.0

IBM. What is supply chain management? *IBM*. Available at: https://www.ibm.com/topics/supply-chain-management

Janssen T. 2017. OOP Concept for Beginners: What is Abstraction? *Stackify*. Available at: https://stackify.com/oop-concept-abstraction/

Logistica efficiente. B2B – Business to Business. Available at: https://www.logisticaefficiente.it/wiki-logistica/supply-chain/b2b-business-to-business.html

Lorenz M., Rüßmann M., Waldner M., Engel P., Harnisch M., Gerbert P., Jan Justus J. 2015. Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries. Available at: https://www.bcg.com/it-it/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries

Lucidchart. UML Activity Diagram Tutorial. Available at: https://www.lucidchart.com/pages/uml-activity-diagram

Maloney A. 2019. The difference between a simulation and a digital twin. *Siemens.* Available at: https://blogs.sw.siemens.com/mindsphere/the-difference-between-a-simulation-and-a-digital-twin/

MathWorks. Triangular distribution. Available at: https://www.mathworks.com/help/stats/triangular-distribution.html

Mecalux. 2019. Supply chain: cos'è e come funziona la catena di approvvigionamento. *Mecalux*. Available at: https://www.mecalux.it/blog/supply-chain-cos-e

Mills G. 2020. Understanding Conveyor Systems. *Thomasnet*. Available at: https://www.thomasnet.com/articles/materials-handling/understanding-conveyor-systems/

Monroy C. 2020. What is a conveyor system? *6 river systems*. Available at: https://6river.com/what-is-a-conveyor-system/

Murray M. 2018. Amazon's Strategy for Pick to Light Systems in Warehouses. *The Balance Small Business.* Available at: https://www.thebalancesmb.com/pick-to-light-warehouse-systems-2221456

Perkins B, Wailgum T. 2017, What is supply chain management (SCM)? Mastering logistics end to end. *CIO*. Available at: https://www.cio.com/article/2439493/what-is-supply-chain-management-scm-mastering-logistics-end-to-end.html

Simulation Paradigms, *CK-12*. Available at: https://www.ck12.org/book/ck-12-modeling-and-simulation-for-high-school-teachers%3a-principles-problems-and-lesson-plans/section/2.3/

Smartdarw. *Activity diagram.* Available at: https://www.smartdraw.com/activity-diagram/

System Dynamics Society. What is SD? *System Dynamics Society*. Available at: https://www.systemdynamics.org/what-is-sd

Taglic N. 2020. What are automated guided vehicles? *6 river systems.* Available at: https://6river.com/what-are-automated-guided-vehicles/

Taliaferro A., Geunette C., Agarwal A., Pochon M. 2016. Industry 4.0 and distribution centers. *Deloitte*. Available at: https://www2.deloitte.com/us/en/insights/focus/industry-4-0/warehousing-distributed-center-operations.html

Tarver E. Value Chain vs. Supply Chain: What's the Difference? *Investopedia.* Available at: https://www.investopedia.com/ask/answers/043015/what-difference-between-value-chain-and-supply-chain.asp

The AnyLogic Company. Available at: https://the.anylogic.company/

UML diagrams. UML Activity Diagram Reference. Available at: https://www.uml-diagrams.org/activity-diagrams-reference.html

WolframMathWorld. Triangular distribution. Available at: https://mathworld.wolfram.com/TriangularDistribution.html

Attachment I - Functions

navigate

```
viewArea.navigateTo();
groupMainMenu.setPos(viewArea.getX(), viewArea.getY());
```

getFreeCell

```
int _rack = 0;
boolean found = false;
if (box.content == empty)
	{
		box.rackLocation = palletRack2.getFreeCell(true);
		_rack = 2;
	}
else if (box.content == kitToStore)
	{
		box.rackLocation = palletRack2.getFreeCell(false);
		_rack = 2;
	}
else if (box.content == products)
	{
		box.rackLocation = palletRack1.getFreeCell(false);
		_rack = 1;
	}
else
	{
		box.rackLocation = palletRack1.getFreeCell(true);
		_rack = 1;
	}
int _row = box.rackLocation.row;
int _position = box.rackLocation.position;
int _level = box.rackLocation.level;
box.row = box.rackLocation.row;
box.position = box.rackLocation.position;
box.level = box.rackLocation.level;
box.rack = _rack;
if (_rack == 1)
{
	palletRack1.reserve(_row, _position, _level, true);
	box.target  =  palletRack1.getPositionAtCellEntry(_row,  _position,  _level,
true, box.target);
}
else
{
	palletRack2.reserve(_row, _position, _level, true);
	box.target  =  palletRack2.getPositionAtCellEntry(_row,  _position,  _level,
true, box.target);
}
```

supplyOrder

```
supplyCompleted = false;
int supplyComponent = (int) selectFrom(components_sheet)
	.where(components_sheet.type.eq(typeSupply))
	.firstResult(components_sheet.batch_size);
int maxContent = (int) selectFrom(components_sheet)
	.where(components_sheet.type.eq(typeSupply))
	.firstResult(components_sheet.max_content_box);
if (typeSupply == C1) {
	c1.inject(supplyComponent);
	box1.inject(supplyComponent/maxContent);
}
else if (typeSupply == C2) {
	c2.inject(supplyComponent);
	box2.inject(supplyComponent/maxContent);
}
else if (typeSupply == C3) {
	c3.inject(supplyComponent);
	box3.inject(supplyComponent/maxContent);
}
```

quantityDropoff

```
if (_box.componentsType == C1) {
	return (int) selectFrom(products_sheet)
			.where(products_sheet.type.eq(kittingBox.productType))
			.firstResult(products_sheet.number_c1);
	}
else if (_box.componentsType == C2) {
		return (int) selectFrom(products_sheet)
				.where(products_sheet.type.eq(kittingBox.productType))
				.firstResult(products_sheet.number_c2);
	}
else { return (int) selectFrom(products_sheet)
				.where(products_sheet.type.eq(kittingBox.productType))
				.firstResult(products_sheet.number_c3);
	}
```

quantityPickup

```
int quantityInBox = 0;
int maxPickup;
maxPickup = (int) selectFrom(components_sheet)
			.where(components_sheet.type.eq(box.componentsType))
			.uniqueResult(components_sheet.max_tote);
quantityInBox = _box.nItems;
return (maxPickup - quantityInBox);
```

selectBox

```
int i=0;
boolean found = false;
while (i < waitTote.size() && found == false){
	if (waitTote.get(i).componentsType == _box.componentsType)
		found = true;
	else i++;
	}
if (found)
	waitTote.free(waitTote.get(i));
else waitTote.free(waitTote.get(0));
```

stockControl

```
int opC1 = (int) selectFrom(components_sheet)
			.where(components_sheet.type.eq(C1))
			.firstResult(components_sheet.order_point);
int opC2 = (int) selectFrom(components_sheet)
			.where(components_sheet.type.eq(C2))
			.firstResult(components_sheet.order_point);
int opC3 = (int) selectFrom(components_sheet)
			.where(components_sheet.type.eq(C3))
			.firstResult(components_sheet.order_point);
if (stockC1 < opC1)
	supplyOrder(C1);
if (stockC2 < opC2)
	supplyOrder(C2);
if (stockC3 < opC3)
	supplyOrder(C3);

if ((stockC1 < opC1) || (stockC2 < opC2) || (stockC3 < opC3))
	stockComplete = false;
else
	stockComplete = true;
```

flowControl

```
if (seizeTransporter.size() + seizeTransporter1.size() + agvOut.size() +
conveyToStation.size() >= 2) {
	hold18.block();
	crane.hold1.block();}
else {
	hold18.unblock();
	crane.hold1.unblock();}
if ((palletRack1.size() < palletRack1.capacity() - 3) && (palletRack2.size() <
palletRack2.capacity() - 3))
	hold19.unblock();
else hold19.block();
if (palletRack2.size() < palletRack2.capacity() - 3){
	hold20.unblock();
	hold21.unblock();}
else {
	hold20.block();
```

```
	hold21.block();}
if (palletRack1.size() < palletRack1.capacity() - 3)
	hold22.unblock();
else hold22.block();
if (stockComplete)
	crane.hold3.unblock();
else
	crane.hold3.block();
```

checkStoredKit

```
int i = 0;
boolean found = false;
Tote _box = new Tote();
Order order = queueAssembly.get(0);
ProductType _product = queueAssembly.get(0).product;
if (kitStored.size() > 0)
{
	while (found == false && i < kitStored.size()) {
		_box = kitStored.get(i);

	     if(_box.productType == _product && _box.reserved == false){
			found = true;
			_box.reserved = true;}
		else i++;
	}
	if (found){
		_box.order = order;
		crane.retrievalList.add(_box);
		crane.enterRetrieval.take(new  Task(false,  true,  1,  taskAssembly,
false));
		}
	else {
		if (stockComplete && kitCompleted)
			kittingOrder(kitToAssemble);
		}
else {
	if (stockComplete && kitCompleted)
		kittingOrder(kitToAssemble);
	}
```

selectStoredKit

```
int _row;
int _position;
int _level;
int i = 0;
boolean found = false;
Tote _box = crane.retrievalList.get(0);
_box.rackLocation = rackSystem.getCellOf(_box);
_row = _box.rackLocation.row;
_position = _box.rackLocation.position;
_level = _box.rackLocation.level;
_box.row = _box.rackLocation.row;
```

```
_box.position = _box.rackLocation.position;
_box.level = _box.rackLocation.level;
_box.process = assembly;
_box.content = kitToAssemble;
_box.targetX = crane.xOutput;
_box.targetZ = crane.arm.zConveyor;
_box.target  =  rackSystem.getPositionAtCellEntry(_row,  _position,  _level,  true,
_box.target);
crane.boxRetrieval = _box;
crane.arm.boxRetrieval = _box;
transferringAssembly.add(queueAssembly.get(0));
queueAssembly.remove(0);
send("transferring", _box.order);
crane.retrievalList.remove(0);
```

kittingOrder

```
boolean foundC1 = false;
boolean foundC2 = false;
boolean foundC3 = false;
int counter = 0;
if (kitCompleted && emptyTotes.size() > 0) {

	kitCompleted = false;
	if (_typology == kitToAssemble)
		crane.enterRetrieval.take(new Task(false, true, 2, taskEmpty, false));
	else crane.enterRetrieval.take(new Task(false, true, 2, taskEmpty1, false));
}
if (emptyChute && stockComplete){
	emptyChute = false;
	for (int w = 0; w < waitChute.size(); w++) {
		if (waitChute.get(w).componentsType == C1)
			foundC1 = true;
		else if (waitChute.get(w).componentsType == C2)
			foundC2 = true;
		else foundC3 = true;
		}
	if (!foundC1)
		counter++;
	if (!foundC2)
		counter++;
	if (!foundC3)
		counter++;
	for (int j = 0; j < counter; j++) {
	crane.enterRetrieval.take(new Task(false, true, 1, taskKitting, false));
	}
}
```

selectEmptyBox

```
int _row;
int _position;
int _level;
int index;
Tote _box = new Tote();
index = uniform_discr(0, emptyTotes.size()-1);
_box = emptyTotes.get(index);
_box.rackLocation = rackSystem.getCellOf(_box);
_row = _box.rackLocation.row;
_position = _box.rackLocation.position;
_level = _box.rackLocation.level;
_box.row = _box.rackLocation.row;
_box.position = _box.rackLocation.position;
_box.level = _box.rackLocation.level;
_box.process = kitting;
_box.content = _content;
_box.targetZ = crane.arm.zConveyor;
_box.targetX = crane.xOutput;
_box.target  =  rackSystem.getPositionAtCellEntry(_row,  _position,  _level,  true,
_box.target);
crane.boxRetrieval = _box;
crane.arm.boxRetrieval = _box;
if (_content == kitToAssemble) {
	_box.order = queueAssembly.get(0);
	_box.productType = queueAssembly.get(0).product;
	transferringAssembly.add(queueAssembly.get(0));
	queueAssembly.remove(0);
	Order newKitOrder = add_orders(kitting, _box.productType);
	transferringKitting.add(0, newKitOrder);
	}
else {
	_box.order = queueKitting.get(0);
	_box.productType = queueKitting.get(0).product;
	transferringKitting.add(queueKitting.get(0));
	queueKitting.remove(0); }

send("transferring", _box.order);
if (_content == kitToAssemble)
	send("transferring", transferringKitting.get(0));
```

selectKittingBox

```
int _row = 0;
int _position = 0;
int _level = 0;
int i = 0;
boolean found = false;
boolean foundC1 = false;
boolean foundC2 = false;
boolean foundC3 = false;
Tote _box = new Tote();
ComponentType component = null;
for (int w = 0; w < waitChute.size(); w++) {
```

```
        if (waitChute.get(w).componentsType == C1)
              foundC1 = true;
        else if (waitChute.get(w).componentsType == C2)
              foundC2 = true;
        else foundC3 = true;
}
if (!foundC1)
        component = C1;
if (!foundC2)
        component = C2;
if (!foundC3)
        component = C3;
while (found == false && i < materialStored.size()) {
        _box = materialStored.get(i);
        if(_box.componentsType == component)
              found = true;
        else
              i++;
}
_box.rackLocation = rackSystem.getCellOf(_box);
_row = _box.rackLocation.row;
_position = _box.rackLocation.position;
_level = _box.rackLocation.level;
_box.row = _box.rackLocation.row;
_box.position = _box.rackLocation.position;
_box.level = _box.rackLocation.level;
_box.target  =  rackSystem.getPositionAtCellEntry(_row,  _position,  _level,  true,
_box.target);
_box.process = chute;
_box.targetZ = crane.arm.zKitting;
_box.targetX = crane.xKitting;
crane.boxRetrieval = _box;
crane.arm.boxRetrieval = _box;
```

checkStoredProduct

```
int i = 0;
boolean found = false;
Tote _box = new Tote();
Order order = queuePicking.get(0);
ProductType _product = queuePicking.get(0).product;
if (productStored.size() > 0)
{
        while (found == false && i < productStored.size()) {
              _box = productStored.get(i);
             if(_box.productType == _product && _box.reserved == false){
                    found = true;
                    _box.reserved = true;}
              else i++;
        }
        if (found) {
              _box.order = order;
              crane.retrievalList.add(_box);
              crane.enterRetrieval.take(new   Task(false,   true,   1,   taskPicking,
false));
```

```
            }
        else
            {
                Order newAssembleOrder = add_orders(assembly, _product);
                queueAssembly.add(0, newAssembleOrder);
                if (transferringAssembly.size()<3)
                    checkStoredKit();
            }
}
else {
            Order newAssembleOrder = add_orders(assembly, _product);
            queueAssembly.add(0, newAssembleOrder);
            if (transferringAssembly.size()<3)
                checkStoredKit();
        }
```

selectPickingBox

```
int _row;
int _position;
int _level;
int i = 0;
boolean found = false;
Tote _box = crane.retrievalList.get(0);
_box.rackLocation = rackSystem.getCellOf(_box);
_row = _box.rackLocation.row;
_position = _box.rackLocation.position;
_level = _box.rackLocation.level;
_box.row = _box.rackLocation.row;
_box.position = _box.rackLocation.position;
_box.level = _box.rackLocation.level;
_box.process = picking;
_box.targetX = crane.xPicking;
_box.targetZ = crane.arm.zPicking;
_box.target  =  rackSystem.getPositionAtCellEntry(_row,  _position,  _level,  true,
_box.target);
crane.boxRetrieval = _box;
crane.arm.boxRetrieval = _box;
transferringPicking.add(queuePicking.get(0));
queuePicking.remove(0);
send("transferring", _box.order);
crane.retrievalList.remove(0);
```

selectAssemblyBox

```
boolean found = false;
int i = 0;
while (i < toteTable.size() && found == false) {
        if (toteTable.get(i).productType == _product.type)
            found = true;
        else i++;
        }
toteTable.free(toteTable.get(i));
```

assignAssemblyBox

```
boolean foundP1 = false;
boolean foundP2 = false;
boolean foundP3 = false;
for (int i = 0; i < toteTable.size(); i++) {
	if (toteTable.get(i).productType == P1)
		foundP1 = true;
	else if (toteTable.get(i).productType == P2)
		foundP2 = true;
	else foundP3 = true;
}
if (!foundP1)
	tote.productType = P1;
if (!foundP2)
	tote.productType = P2;
if (!foundP3)
	tote.productType = P3;
tote.content = empty;
tote.process = storage;
```

schedulePickingOrder

```
int i=1;
boolean found = false;

while (found == false && i < queuePicking.size()) {

	for (int k=0; k < productStored.size(); k++) {
		if (queuePicking.get(i).product == productStored.get(k).productType)
			found = true;
		}

	i++;

	}

i = i-1;

if (found) {
	Order orderToSchedule = queuePicking.get(i);
	queuePicking.remove(i);
	queuePicking.add(0, orderToSchedule);
	scheduled++;
	checkStoredProduct();
	}
```

Attachment II – Experiments results

Experiment 1 - Base model

EXPERIMENT 1 - BASE MODEL							
CATEGORY	**KPI**	**TEST 1**	**TEST 2**	**TEST 3**	**TEST 4**	**TEST 5**	**AVERAGE**
Utilization [%]	AGVs	0.18	0.14	0.1	0.13	0.17	14.4%
	Worker 1	0.78	0.6	0.53	0.6	0.78	65.8%
	Worker 2	0.15	0.13	0.13	0.13	0.15	13.8%
	Worker 3	-	-	-	-	-	-
	Mini-load	0.16	0.14	0.12	0.14	0.17	14.6%
	Receiving station	0.02	0.02	0.58	0.01	0.02	13.0%
	Kitting station	0.06	0.05	0.04	0.05	0.07	5.4%
	Picking station	0.01	0.02	0.03	0.02	0.02	2.0%
	Assembly station 1	0.75	0.61	0.51	0.61	0.75	64.6%
	Assembly station 2	-	-	-	-	-	-
	Rack	0.42	0.48	0.33	0.5	0.48	44.2%
Mini-load [s]	Queue time	1.12	1.47	1.15	1.35	2.04	1.43
	Execution time	6.57	6.57	6.57	6.59	6.56	6.57
	Cycle time	7.69	8.04	7.72	7.93	8.61	8.00
	Storage lead time	10.78	11.16	10.66	11.04	11.15	10.96
	Retrieval lead time	8.58	9.49	8.62	9.37	10.848	9.38
	Transfer lead time	12.93	12.2	12.56	12.22	13.6	12.70
Kitting [s]	Kitting waiting	11.13	1.56	0.76	1.57	12.83	5.57
	Kitting transferring	13	12.85	14.12	12.95	12.79	13.14
	Kitting processing	11.61	11.66	11.47	11.73	11.73	11.64
	Kitting cycle time	24.61	24.51	25.59	24.68	24.52	24.78
	Kitting order time	35.74	26.07	26.36	26.25	37.35	30.35
Assembly [s]	Assembly waiting	52.96	47.81	51.58	47.81	32.24	46.48
	Assembly transferring	257.84	104.49	140.06	104.53	262.75	173.93
	Assembly processing	232.97	234.8	229.79	234.8	230.9	232.65
	Assembly cycle time	490.81	339.29	369.85	339.32	493.65	406.58
	Assembly order time	543.78	387.1	421.43	387.12	525.89	453.06
Picking [s]	Picking waiting	0	0	0.56	0	0	0.11
	Picking transferring	12.49	15.41	12.19	15.41	8.75	12.85
	Picking processing	13.55	14.09	14.07	14.09	13.89	13.94
	Picking cycle time	26.04	29.5	26.26	29.5	22.65	26.79
	Picking order time	26.04	29.5	26.82	29.5	22.65	26.90
Average WIP [n° of items]	Mini-load	0.19	0.17	0.15	0.17	0.2	0.18
	Kitting	0.18	0.12	0.09	0.12	0.21	0.14
	Assembly	2.21	0.97	1	0.97	2.12	1.45
	Picking	0.03	0.09	0.06	0.06	0.03	0.05
Throughput per hour [ops/hour]	Mini-load	78.56	70.93	71.13	70.84	80.17	74.33
	Kitting	13.1	10.85	10.13	10.85	13.59	11.70
	Assembly	6.84	6.29	5.81	6.29	6.83	6.41
	Picking	3.25	3.33	4.62	3.33	3.32	3.57

Experiment 2 – Modified rates

EXPERIMENT 2 - MODIFIED DEMAND RATES							
CATEGORY	**KPI**	**TEST 1**	**TEST 2**	**TEST 3**	**TEST 4**	**TEST 5**	**AVERAGE**
Utilization [%]	AGVs	0.22	0.2	0.16	0.2	0.25	20.6%
	Worker 1	0.78	0.87	0.63	0.86	0.76	78.0%
	Worker 2	0.27	0.26	0.23	0.24	0.26	25.2%
	Worker 3	-	-	-	-	-	-
	Mini-load	0.24	0.23	0.19	0.23	0.25	22.8%
	Receiving station	0.02	0.02	0.02	0.02	0.02	2.0%
	Kitting station	0.08	0.08	0.06	0.07	0.09	7.6%
	Picking station	0.07	0.07	0.08	0.06	0.1	7.6%
	Assembly station 1	0.75	0.83	0.6	0.83	0.73	74.8%
	Assembly station 2	-	-	-	-	-	-
	Rack	0.52	0.44	0.42	0.48	0.54	48.0%
Mini-load [s]	Queue time	1.89	1.5	1.27	1.36	1.72	1.55
	Execution time	6.56	6.56	6.55	6.57	6.56	6.56
	Cycle time	8.45	8.05	7.83	7.93	8.28	8.11
	Storage lead time	11.01	11.05	10.72	11.22	11.2	11.04
	Retrieval lead time	10.18	9.2	8.91	8.54	9.77	9.32
	Transfer lead time	12.76	12.36	12.28	13.12	12.87	12.68
Kitting [s]	Kitting waiting	39.91	4.84	4.84	85.24	33.32	33.63
	Kitting transferring	14.1	15.29	13.93	13.45	13.52	14.06
	Kitting processing	11.88	11.86	11.86	11.47	11.55	11.72
	Kitting cycle time	25.98	27.15	25.79	24.91	25.07	25.78
	Kitting order time	65.89	31.99	30.63	110.16	58.39	59.41
Assembly [s]	Assembly waiting	32.97	53.94	28.85	117.26	21.42	50.89
	Assembly transferring	344.87	346.18	243.63	378.02	352.51	333.04
	Assembly processing	241.13	234.54	230.24	237.88	226.5	234.06
	Assembly cycle time	586	580.73	473.88	615.9	579.01	567.10
	Assembly order time	618.97	634.67	502.72	733.16	600.42	617.99
Picking [s]	Picking waiting	10.05	21.52	0.73	0.66	0	6.59
	Picking transferring	11.21	14.23	15.04	11.17	10.63	12.46
	Picking processing	13.97	14.01	13.96	14.09	13.98	14.00
	Picking cycle time	25.18	28.24	29	25.26	24.61	26.46
	Picking order time	35.23	49.76	29.73	25.93	24.61	33.05
Average WIP [n° of items]	Mini-load	0.29	0.28	0.24	0.28	0.29	0.28
	Kitting	0.45	0.2	0.16	0.74	0.47	0.40
	Assembly	2.58	3.57	1.45	2.95	4.1	2.93
	Picking	0.7	0.86	0.68	0.09	3.48	1.16
Throughput per hour [ops/hour]	Mini-load	111.03	122.36	107.57	129.62	127.87	119.69
	Kitting	16.13	18.97	14.61	20.53	20.64	18.18
	Assembly	7.04	8.28	6.31	7.93	7.46	7.40
	Picking	12.65	13.02	13.62	14.15	15.16	13.72

Experiment 3 – Modified layout

EXPERIMENT 3 - MODIFIED LAYOUT							
CATEGORY	**KPI**	**TEST 1**	**TEST 2**	**TEST 3**	**TEST 4**	**TEST 5**	**AVERAGE**
Utilization [%]	AGVs	0.24	0.21	0.24	0.2	0.23	22.4%
	Worker 1	0.7	0.57	0.67	0.4	0.53	57.4%
	Worker 2	0.32	0.27	0.25	0.31	0.26	28.2%
	Worker 3	0.35	0.5	0.57	0.24	0.41	41.4%
	Mini-load	0.31	0.24	0.27	0.26	0.27	27.0%
	Receiving station	0.03	0.02	0.03	0.03	0	2.2%
	Kitting station	0.12	0.07	0.1	0.08	0.09	9.2%
	Picking station	0.08	0.07	0.06	0.08	0.09	7.6%
	Assembly station 1	0.64	0.5	0.61	0.35	0.48	51.6%
	Assembly station 2	0.32	0.48	0.52	0.21	0.37	38.0%
	Rack	0.58	0.44	0.56	0.65	0.48	54.2%
Mini-load [s]	Queue time	2.13	1.69	2.11	2.13	2.21	2.05
	Execution time	6.57	6.57	6.58	6.56	6.53	6.56
	Cycle time	8.7	8.26	8.69	8.69	8.74	8.62
	Storage lead time	12.2	11.41	12.99	12.1	12.56	12.25
	Retrieval lead time	9.87	9.3	9.38	9.79	9.48	9.56
	Transfer lead time	12.22	13.52	12.91	12.21	15.75	13.32
Kitting [s]	Kitting waiting	40.09	7.92	33.89	8.24	31.88	24.40
	Kitting transferring	16.47	15.72	16.16	16.32	14.33	15.80
	Kitting processing	11.8	11.59	11.59	11.58	11.64	11.64
	Kitting cycle time	28.27	27.31	27.75	27.9	25.96	27.44
	Kitting order time	68.36	35.23	61.65	36.14	57.84	51.84
Assembly [s]	Assembly waiting	25.77	2.41	5.1	21.39	45.79	20.09
	Assembly transferring	68.85	59.89	103.55	37.64	190.41	92.07
	Assembly processing	240.3	242.12	244.44	249.17	249.03	245.01
	Assembly cycle time	309.15	302.02	347.99	286.8	439.44	337.08
	Assembly order time	334.93	304.42	353.09	308.19	485.22	357.17
Picking [s]	Picking waiting	30.27	18.51	46.9	59.88	16.75	34.46
	Picking transferring	17.21	13.64	13.18	15.02	8.85	13.58
	Picking processing	13.81	14.11	14.12	14.09	14.15	14.06
	Picking cycle time	31.02	27.75	27.39	29.11	23.01	27.66
	Picking order time	62.29	46.26	74.29	88.99	39.76	62.32
Average WIP [n° of items]	Mini-load	0.46	0.34	0.39	0.38	0.38	0.39
	Kitting	0.63	0.22	0.49	0.25	0.42	0.40
	Assembly	1.4	1.31	1.81	0.76	2.41	1.54
	Picking	0.63	0.75	1.89	0.51	2.37	1.23
Throughput per hour [ops/hour]	Mini-load	161.77	137.93	143.08	142.76	136.02	144.31
	Kitting	26.96	21.81	23.89	22.25	22.57	23.50
	Assembly	8.65	8.54	9.03	6.89	5.65	7.75
	Picking	15.39	15.16	13.45	16.48	15.44	15.18

Experiment 4 – Modified priorities

EXPERIMENT 4 - MODIFIED PRIORITIES							
CATEGORY	**KPI**	**TEST 1**	**TEST 2**	**TEST 3**	**TEST 4**	**TEST 5**	**AVERAGE**
Utilization [%]	AGVs	0.26	0.2	0.23	0.26	0.31	25.2%
	Worker 1	0.79	0.69	0.65	0.84	0.84	76.2%
	Worker 2	0.33	0.25	0.31	0.33	0.34	31.2%
	Worker 3	0.62	0.5	0.45	0.73	0.72	60.4%
	Mini-load	0.29	0.26	0.26	0.28	0.32	28.2%
	Receiving station	0.02	0.02	0.02	0.02	0.02	2.0%
	Kitting station	0.1	0.08	0.09	0	0.11	7.6%
	Picking station	0.08	0.08	0.09	0.1	0.07	8.4%
	Assembly station 1	0.72	0.63	0.59	0.76	0.78	69.6%
	Assembly station 2	0.58	0.7	0.42	0.7	0.69	61.8%
	Rack	0.35	0.38	0.44	0.4	0.35	38.4%
Mini-load [s]	Queue time	1.99	1.92	1.71	1.67	2.1	1.88
	Execution time	6.61	6.56	6.57	6.58	6.58	6.58
	Cycle time	8.59	8.48	8.28	8.24	8.69	8.46
	Storage lead time	11.55	11.2	11.23	11.14	12.51	11.53
	Retrieval lead time	9.44	9.66	9.32	8.71	9.14	9.25
	Transfer lead time	14.53	14.74	14.34	15.06	14.33	14.60
Kitting [s]	Kitting waiting	28.44	0	0.14	35	32.35	19.19
	Kitting transferring	15.77	14.35	16.59	16.17	15.1	15.60
	Kitting processing	11.56	11.54	11.71	11.69	11.43	11.59
	Kitting cycle time	27.32	25.89	28.3	27.86	26.53	27.18
	Kitting order time	55.76	25.89	28.45	62.86	58.88	46.37
Assembly [s]	Assembly waiting	18.56	10.72	5.92	5.21	31.14	14.31
	Assembly transferring	236.71	229.92	120.81	200.4	252.1	207.99
	Assembly processing	247.32	240.51	245.54	241.41	242.25	243.41
	Assembly cycle time	484.02	470.43	366.35	441.81	494.35	451.39
	Assembly order time	502.58	481.15	372.27	447.02	525.49	465.70
Picking [s]	Picking waiting	12.28	16.73	12.68	11.48	106.21	31.88
	Picking transferring	11.82	12.48	11.45	13.67	10.49	11.98
	Picking processing	14.28	14.44	14.09	14.32	14.2	14.27
	Picking cycle time	26.1	26.91	25.54	27.98	24.69	26.24
	Picking order time	38.38	43.64	38.23	39.46	130.9	58.12
Average WIP [n° of items]	Mini-load	0.4	0.32	0.36	0.38	0.44	0.38
	Kitting	0.46	0.16	0.21	0.51	0.56	0.38
	Assembly	3.8	2.39	1.85	2.93	4.49	3.09
	Picking	1.29	1	0.43	1.23	2.45	1.28
Throughput per hour [ops/hour]	Mini-load	148.26	129.81	131.85	145	152.5	141.48
	Kitting	20.07	17.89	18	18.7	22.2	19.37
	Assembly	10.56	8.03	8.15	11.87	11.98	10.12
	Picking	24.1	19.18	20.14	24.91	22.01	22.07