Simulation and Digital Twin: Contrasting the Capabilities and Bridging the Gap

by

Ana Carolina Wooley

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 5, 2023

Keywords: simulation, digital twin, capability, manufacturing

Approved by

Daniel F. Silva, Chair, Assistant Professor of Industrial and Systems Engineering
Gregory Harris, Associate Professor of Industrial and Systems Engineering
Gregory Purdy, Assistant Professor of Industrial and Systems Engineering
Konstantinos Mykoniatis, Assistant Professor of Industrial and Systems Engineering

Abstract

Driven by the advancements related to Industry 4.0, Digital Twin (DT), usually described as a virtual representation of a physical product or system connected with bi-directional data, has been a topic of increasing interest by both academia and industry. As a consequence of this growing interest, many works have been published about DT and its applications, leading to a dilution of the concept of DT, which is engendering a misunderstanding of the application of DT and its benefits. The use of simulation combined with a DT is quite common, which creates an immense misconception about classifying a simulation model as a DT and vice versa. In fact, several papers reviewed herein build simulations and call them DT, but it is unclear if these simulations have the full capabilities usually associated with DT. Simulation and DT are distinct technologies with unique benefits; therefore, they should be classified accordingly, and it is crucial to clarify their differences to prevent misunderstandings and achieve consistency in DT implementations. Therefore, in order to fill this gap, this dissertation explores the levels of capability's levels of both simulation and DT, investigating the connection between them and demonstrating how to bridge this gap. The first contribution of this work is to propose a simulation capability framework and provide an example of a fully capable simulation application, adding and illustrating each level of capability, one at a time. This proposed framework, later referred to as the 4S framework, is different from the existing frameworks for simulation because it is directly comparable and analogous to an existing framework for building DTs, called the 4R framework. In the second contribution, a systematic literature review was performed to investigate whether the current literature is truly applying DT or using a simulation in its place. In this contribution, the 4R framework and the 4S framework provided in the first contribution are used to classify the works herein as either simulation and/or DT, based on the capabilities that they have. In the third contribution, a real-world case study that documents the process of transforming a fully capable DES into the first two levels of capability of a DT is presented. This contribution identifies the key steps involved in this transformation, documents the challenges encountered, and present the solutions that were found. The finding of this work and its future extensions can potentially help academia and the industry by moving the discussion towards a consensus about when a simulation model is a DT and when it is not.

Acknowledgements

I would like to express my appreciation to my advisor, Dr. Daniel F. Silva, for his constant support and guidance throughout my Ph.D. journey. I am truly fortunate to have had such a dedicated and knowledgeable mentor. Thank you for being such a wonderful example of a professor who cares deeply for his students' success. You are my forever mentor!

I would also like to extend my sincere appreciation to the members of my committee, Dr. Gregory Harris, Dr. Gregory Purdy, and Dr. Konstantinos Mykoniatis. Your valuable insights and constructive feedback have been instrumental in refining my work and broadening my perspectives. Thank you for the time and effort you devoted to reviewing my progress and providing suggestions. Also, I would like to thank Dr. Jeff Smith for his assistance and dedication to supporting me in building simulation models and Dr. SueAnne Griffith for graciously agreeing to serve as the university reader.

A special mention goes to my dear friend, Julia Bitencourt, who was a constant companion on this journey. Your presence, brainstorming sessions, and constant support have been instrumental in overcoming challenges. Miga, thank you so much for being there for me! I would also like to acknowledge the contributions and support of all my friends who have helped and encouraged me throughout this journey.

I am deeply grateful to the automotive manufacturing plant that graciously opened its doors, allowing me to gain a deep understanding of its system and develop simulation models of their plant. Shaine Stewart and Cody Cline, your willingness to collaborate and share your expertise has been truly invaluable.

Finally, I would like to express my deepest gratitude to my family, especially my husband, Aaron Wooley, for their unwavering love, support, and understanding. Your encouragement and

belief in me have been the driving force behind my achievements. Love you all! To my dear dad, who is no longer with us, I know he would have been immensely proud of this accomplishment. Love you forever, Dad!

This work would not have been possible without the collective support, guidance, and encouragement of all those mentioned above, as well as the many others who have contributed to my academic and personal growth. Thank you all for being an integral part of my journey as I strive to follow God's plan for my life!

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
AEL	Alternative Elevator Logic
AIC	Akaike information criterion
BL	Baseline Model
CNC	Computer Numerical Control
DES	Discrete Event Simulation
DC	Die Cast
DT	Digital Twin
IoT	Internet of Things
L1	Layout 1
L2	Layout 2
MC	Machining
ML	Machine Learning
NASA	National Aeronautics and Space Administration
PLC	Programmable Logic Controllers
R1	Representation
R2	Replication
R3	Reality
R4	Relational
RQ	Research Question
SLR	Systematic Literature Review
S1	Modeling

S2	Analyzing
S3	Predicting
S4	Prescribing
VR	Virtual Reality
VSM	Value Stream Mapping
WIP	Work in Process
WSC	Winter Simulation Conference

Chapter 1

Introduction

1.1 Background

The fourth industrial revolution, known as Industry 4.0, is characterized by a range of new technologies that are promoting the digitalization of industrial processes by fusing the physical and digital worlds [1,2]. One important component of this is the Digital Twin (DT) [3–6], which is composed of three elements: physical space, virtual space, and the data flow that links the physical and virtual spaces in both directions. DT has the potential to improve a manufacturing system by predicting future events, based on its current status and expected behavior, and then implementing the necessary responses to optimize performance in near-real-time. This is made possible by the growth in internet of things (IoT) connectivity, computing power, and advanced analytics [7].

Simulation has been defined as the imitation of the operation of a real-world process or system over time [8]. Simulation has been extensively utilized for evaluating and optimizing the design and operation of production systems [9] and DT has been referred to as the newest wave in simulation technology [10]. In the manufacturing domain, the most popular simulation technique is discrete event simulation (DES) [11]. A DES model is composed of interacting entities that represent the tangible (such as a manufacturing station) and intangible (such as a queue of work-in-progress) components of the system being modeled and are updated at discrete but potentially random event times [12]. A simulation model and a DT are both digital representations of a physical system [13,14]; however, they have different capabilities, which will be clarified in this research.

Further research on the connection between the concepts of simulation and DT is needed [15], and most of researchers do not usually distinguish DT from a general simulation model [16]. To fill this gap, this research focuses on exploring the levels of capability of both simulation and DT and demonstrating how to bridge this gap.

1.2 Research Objectives

The ability to replicate a physical system in a virtual environment is shared by both traditional simulation and DT models,which creates a huge misunderstanding about classifying a simulation model as a DT. It is essential to classify them correctly and to clarify their differences to prevent misunderstandings. The overall objective of this research is to identify the differences between simulation and DT in terms of its capabilities, providing academia and industry with clear paths to application and research. In this research, three gaps that are not currently covered in the literature are identified and three contributions are proposed to fill these gaps:

- The first contribution of this research is to define the levels of capability of a simulation model into a 4S framework, that is analogous to the 4R framework for DTs, mentioned in Section 2.5.1. In this contribution, an example is provided of a successful, fully capable simulation application created from a VSM, which is based on the proposed 4S framework. This novel approach demonstrates how to add capability levels to a simulation model, one at a time.
- The second contribution of this research is to perform a systematic literature review and use to investigate whether researchers are truly applying a DT approach or if they are using simulation in place of a DT. This investigation is performed by using the 4R and 4S

frameworks to classify the works herein as either simulation and/or DT, based on the capabilities presented.

- The third contribution of this research is to bridge the gap between simulation and DT by providing a framework that is intended to guide researchers in building a DT directly from a fully capable simulation and implementing it in an industrial setting.

The expected significance of this research is that these three major contributions can add value by providing a clearer understanding about the differences in capabilities between simulation and DT. This research will help create a sense of comprehension and understanding of these topics. As will be described in Section 2.4, simulation has a wide variety of types. This research is focused on DES since it is the most common type of simulation used in the manufacturing domain.

1.3 Dissertation Organization

This dissertation is organized as follows: Chapter 2 provides a background and literature review that serves as a foundation of the topics related to this dissertation. Chapter 3 outlines the purpose, objectives, and scope of this research. Chapter 4, 5, and 6 introduce the three main contributions of this research. Chapter 4 contains simulation capabilities framework and application of a fully capable simulation. Chapter 5 offers a systematic literature review to classify the DT applications within its capabilities. Chapter 6 proposes a framework to guide the creation of a DT from a fully capable simulation. The dissertation draws to a close with Chapter 7 presenting the conclusions and future work, which is followed by References.

Chapter 2

Background and Literature Review

The purpose of this chapter is to provide the background information that serves as the foundation for each contribution and is necessary to build a case for the need of differentiating simulation from DT in terms of its capabilities.

The fourth industrial revolution and its underlying digital transformation, known as Industry 4.0, is advancing at a rapid pace [17]. Digitalization offers the advantage of enabling the real-time monitoring of the manufacturing space [18]. Through digitalization, monitoring tools can better assist decision makers in efficiently capturing non-value-added processes on the factory floor. Value Stream Mapping (VSM) is a common monitoring tool, but the implementation is challenging when the product processing is more complex and requires improvements in labor management and facility utilization [19]. For that reason, combining simulation and VSM can help manufacturers to fully capture the manufacturing setting, further analyze data, and generate performance statistics without interfering with the real system, enabling decision makers to understand how changes in the process will be reflected throughout the system [19–22].

Although conventional VSM is not adept to deal with today’s fast-paced, dynamic manufacturing environment, complex material flow, or machine and labor performance efficiency, it is still useful as a primary tool to identify sources of waste in the system [23]. After identifying the area that is causing problems in the system, simulation can be used to conduct analyses of multiple scenarios and suggest changes without having to interfere with the real system [21,23]. Moreover, simulation is an excellent tool for accurately representing reality in a virtual environment of a manufacturing system [24]. Because the simulation models are so intricate, running simulation trials might take hours and a vast amount of production data [8]. A better approach is to implement DT, which is an important component of the fourth industrial revolution, and it can simulate the real-time behavior of a physical system, using sensors and information systems to mirror its current condition [3,10]. Similar to simulation, DTs are also used to understand a system and make predictions; however, they are capable of accurately representing the system in real-time and having a bidirectional connection of data [14].

The natural progression to approach a problem in the manufacturing environment, suggested in this dissertation, is to first start with a VSM to identify the main source of waste in the system, then use a DES model to further analyze alternative scenarios, and progress from that to a DT. The complexity, detail, and data increase as it goes from VSM to DES to DT. Overall, the major goal of this dissertation is to provide an example of how one can do this entire process. The case studies and applications provided throughout this dissertation are focused on automotive manufacturing, and the simulation type used is DES since it's the most common type of simulation used in the manufacturing domain.

In this chapter, the concepts of Industry 4.0 (Section 2.2), Value Stream Mapping (Section 2.3), Simulation (Section 2.4), Discrete Event Simulation (Section 2.4.1), Data-Generated Simulation Models (Section 2.4.2), Digital Twin (Section 2.5), and the 4R Framework (Section 2.5.1) are explored. These concepts are evaluated throughout this dissertation, exploring the connection between them. The three main contributions of this dissertation are presented in Chapter 4, 5, and 6, consecutively.

2.2 Industry 4.0

Historically, industrial revolutions have been characterized by technological advancements that involve social and economic progress. The fourth industrial revolution, known as Industry 4.0, has been made possible in recent years due to advancements in technology and science, particularly in areas such as automation, IoT, cloud computing, artificial intelligence (AI), and big data analytics [17]. These technologies have enabled machines, devices, and systems to communicate and interact with each other in real-time, leading to significant improvements in efficiency, productivity, and innovation in various industries [18]. This latest industry revolution relies on the use of digital technologies to promote the digitalization of industrial processes. By implementing information technologies, Industry 4.0 aspires to fully integrate all the parts involved, such as logistics, and suppliers. Therefore, connectivity is required to develop a cooperative technological network to improve efficiency in production [18]. Additionally, Industry 4.0 requires the use of recent technologies such as IoT, Cloud computing, and DT [25]. Real-time visibility of manufacturing assets is ensured by analyzing massive amounts of data gathered from sensors on the factory floor [26].

The concepts and technologies of Industry 4.0 can be applied to many of industrial sectors, including discrete and continuous process manufacturing [27]. Smart factories employing IoT technology have higher production and better quality [28]. Manufacturing errors are decreased, and money and time are saved, when manual inspection business models are replaced with AI powered visual insights. A smartphone connected to the cloud can be easily set up to enable remote monitoring of manufacturing processes [29].

2.3 Value Stream Mapping

Automotive manufacturers are increasing the implementation of lean manufacturing principles and tools, such as continuous flow and VSM, to identify, eliminate or reduce waste and improve the system [22,23,30]. In the context of lean manufacturing, waste can be defined as any activity or process that consumes resources (time, money, materials, etc.) but does not add value to the final product or service from the customer's perspective [20]. VSM is a lean tool that helps to identify and eliminate sources of waste in a manufacturing process to serve customers with higher quality and promptness [20]. The analysis of a VSM leads to identifying the non-value-added steps that exist in a process and the main purpose of applying this tool is to identify waste and opportunities to implement solutions [19].

VSM can be used in conjunction with DES to help automotive organizations to further analyze data, quantify gains, identify needs for resources, and generate performance statistics without interfering with the real system [21]. Both techniques, DES and VSM, have a long history of application in automotive manufacturing. However, they are usually applied separately. A few publications have highlighted the application of VSM together with DES in manufacturing. For instance, [22] combines VSM and DES as a decision-making tool of an assembly line of clutch discs in an automotive company. [31] proposed a method for using simulation to extend the analysis of a VSM, evaluating two scenarios of push and pull manufacturing systems. [32] presented a lean assessment framework that integrates VSM with simulation in a tire distribution company. [33] discussed the combination of VSM with simulation and concluded it is beneficial not only for verification but also for seeing different pictures of the investigated manufacturing system. [21] stated that combining VSM with simulation provides an accurate analysis of system’s current and future states and it is an efficient quantitative assessment for the lean practices and policies.

2.4 Simulation

Simulation is a model that imitates the operation of an existing process or system over time that is used to describe and analyze the behavior of systems as well as to run what-if scenarios [34]. There are many advantages to using simulation. It offers methods and tools to model system behaviors in great detail and use them to predict future outcomes, thereby supporting decision-making processes [35]. Simulation allows testing designs without committing resources to acquisition. It also allows the option to compress and expand time, which is relevant when investigating a system [36]. Simulation also provides understanding of why things happen the way they do in the system and explores possibilities without the expense and disruption of experimenting with the real system [8]. Overall, a simulation model is an accurate representation of a system or subsystem that satisfies a particular set of criteria, and it is often used to verify and analyze potential improvement scenarios before the actual implementation in the factory [37].

Simulations can be deterministic or stochastic. Randomness is incorporated to reflect the uncertainty found in most systems in a stochastic simulation, which is the most popular approach [38]. Models that are deterministic have no variability. These are uncommon in design applications, but they are more typical in model-based decision support applications like scheduling [8,39]. Moreover, a simulation can also be discrete or continuous, which describes how the states of the system change over time. In a discrete simulation, some states, such as the status of a worker, can only change at specific points in time (called event times) [38,40,41]. In a continuous simulation, the states can change continuously throughout time, such as the temperature in an oven. While most systems only have discrete or continuous states, others also contain both kinds of states [39,42].

The use of simulation technology is well known in the field of engineering [43] and there is a large range of simulation types that can be selected [24]. Selecting which type to use depends on the characteristics of the actual system being considered, specific requirements that need to be met, and the desired outcomes of the analysis. The three most common types of simulation are Monte Carlo, Agent-Based, and DES [44]. In simple terms, a Monte Carlo simulation is a method that identifies uncertainties and potential risks through probability distributions [45]. An Agent-Based simulation models individual entities (agents) and their interactions with each other and the environment, which allows for the modeling of complex systems with many interacting parts [46]. DES is the most popular method applied in the manufacturing domain. It allows the modeler to observe specific events that are result from the business processes and it can be used to study many types of systems and for a diverse range of outcomes [37].

There is extensive literature defining simulation and providing step-by-step guides on how to build a simulation model. For instance, [47] specifies multiple outcomes that a simulation model can accomplish, including output analysis and problem diagnosis. [48] defines a simulation method for building a model and conducting experiments on that model. An experiment entails repeatedly running the simulation for a time period in order to obtain data for statistical analysis. Similarly, [49] states that simulation can mimic the dynamics of a system and is able to analyze the system by running different scenarios. Comparably, [50] describes that a simulation model is capable of conducting experiments that provide understanding of the behavior of the system and able to evaluate the effect of different input levels on specified measurements of performance. [8] defines that simulation is used to describe and analyze the behavior of a system and ask what-if questions about the system. Overall, these authors agree that a simulation model is capable of conducting experiments and analyzing results; however, none of them provide a clear framework of simulation capabilities that enables us to distinguish between the different levels of capability, or that can be directly compared to other technologies, such as DT.

2.4.1 Discrete Event Simulation

DES models emulate the real-world system in an event-based sequence, enabling the decision makers to understand and optimize their operations [51]. A DES models a system whose states may change only at a discrete point in time [52]. A system is composed of objects called entities that have certain attributes [53]. A state is a collection of attributes or variables that represent the entity of the system [47]. An event represents an occurrence that changes the state of the system, and each event takes place at a specific moment in time [54]. The state variables change only at those discrete points in time at which events occur, which is a consequence of activity times and delays [8,38]. Hence, a DES model allows companies to make predictions about the behavior of processes, objects, and events in the short and long-term [50,55].

A DES model comprises four main components: system entities, input variables, performance measures, and functional relationships. For instance, a server and a queue are considered systems entities, arrival rate is an example of input variables, performance measures could be the average wait time in queue, and an example of a functional relationship is the time in system because it involves the wait and service times [40]. To develop a DES model, the modeler should first have a thorough understanding of the system being modeled and a fundamental modeling strategy [8,56]. Simulation models are created to achieve particular goals and ensure that the model will provide answers to the desired questions [57].

DES have been traditionally used in industrial applications [58], including automotive manufacturing, because they can incorporate several sources of variability, complex constrains, and interactions that are difficult to measure and understand directly from the real system [59]. Several researchers have applied DES in automotive manufacturing [30]. [60] presented a simulation methodology to model the production floor, warehouse, and material handling system of an automotive assembly facility. [61] demonstrated the importance of using simulation to test what-if scenarios in a manufacturing process. [62] proposed a simulation of an automotive manufacturing plant that can be automatically generated. Manufacturing research is experimentally oriented, and simulation models have long been applied to analyze different experiments and optimize a system through "what-if" analysis [63].

2.4.2 Data-Generated Simulation Models

As previously mentioned, simulation modeling has been extensively employed to assess the intricacies of complex systems, encompassing aspects such as design, operation, and performance [35]; and DES has been a commonly employed approach to model complex systems [34]. While these models are manually constructed and can possess a high level of complexity, their effectiveness in capturing changes within reconfigurable manufacturing systems, which experience continuous modifications in their physical and software structures, is very limited [64]. Additionally, as systems evolve and interact with their environments, they often exhibit new and unforeseen behaviors [65]. Consequently, manually created simulation models quickly become outdated after their development, necessitating constant and costly development of new models or manual updates to existing ones [66].

In contrast to this, data-generated models have the capability to capture these modifications in real-time [67]. The concept of data-generated models revolves around creating generic models where a sizable portion of the data for constructing the models is imported from external files [39]. This approach provides an efficient solution with diverse options for connecting to different data sources, such as direct database binding, and spreadsheet and CSV file binding. Direct database binding enables efficient querying and accessing data stored in database systems, providing a convenient and robust method for data retrieval and utilization [39,68]. Spreadsheet and CSV file binding enables the extraction and utilization of data stored in these formats [69]. It provides flexibility in handling tabular data, facilitating data manipulation, analysis, and integration with the simulation model [39]. The key advantage of data-generated models is that they can adapt and reflect changes as they occur, eliminating the need for constant model redevelopment or updates [70]. Moreover, these models could, at least in theory, be effectively deployed as DTs, playing a crucial role in supporting design, planning, and scheduling processes [39]. This should offer a more efficient and cost-effective solution compared to manually created simulation models [67].

The transition from traditional simulation modeling to a data-generated approach has been supported by several factors. First, data-generated models tend to provide a more precise representation of the systems they model in comparison to traditional methods [71]. Specifically, real-world systems often exhibit unpredictable behaviors that are challenging to anticipate and pre-model. These elusive behaviors can only be effectively captured and incorporated into models through the utilization of real-time data [71]. Additionally, Machine Learning (ML) and AI methods can be integrated with simulation modeling to gain a more comprehensive understanding of complex system behaviors [72]. This combination allows for a deeper exploration and analysis of system dynamics that may not be achievable through standard simulation modeling alone [73].

2.5 Digital Twin

DT has been a topic of increasing interest by both academia and industry [16,74]. For instance, companies such as IBM, Siemens, and General Electric are already adopting DT technologies [75]. This growth is driven by Industry 4.0 and advances in technologies such as the IoT, AI, wireless sensor networks, ML, and big data, providing opportunities to integrate physical and virtual environments [76]. As a consequence of this growing interest, many works have been published about DT and its applications [77,78], leading to a dilution of the concept of DT [79–81], which is engendering a misunderstanding of the application of DT and its benefits [82].

It is acknowledged that the concept of DT was first introduced by Michael Grieves in collaboration with John Vickers from National Aeronautics and Space Administration (NASA) [83]. Grieves defines a DT model as one that is composed of three elements: physical space, virtual space, and the data flow that connects the physical space to the virtual space and vice versa. Since then, researchers have been trying to reach a consensus on a single and unified definition that sufficiently represents a DT [77,81]. For instance, [84–87] consider DT of a product, in which it represents a single machine or process, whereas [4,13,88–90] consider DT of a process, which encompass a section of or the entire production environment.

Several previous works have focused on providing an architecture of DT and showing how the various DT components interact. On the journey to determine the definition and capabilities of DT, several previous related terms have existed, such as: Digital Model and Digital Shadow [13]. Some researchers, such as [91,92], present DT concepts, definitions, and architecture. Others, such as [93,94], explore the trending technologies for industrial DT application. Although some authors describe DT as a model in which integrates the physical system with the virtual counterpart with a constant and bidirectional data flow between them [13] , there are still plenty of definitions that describe DTs as only a true representation of the physical environment, ignoring the real-time data connection [95]. To reach a consensus on a DT definition, the specifications of the fundamental requirements and capabilities for a DT are necessary and these requirements have changed over time because they are linked to advancements in the technologies, such as ML, and big data [95].

2.5.1 The 4R Framework

Definitions of the levels of capability and complexity of a DT are described by [96,97]. They propose a versatile structure for designing and implementing a general-purpose DT and call it the 4R framework. This 4R framework is composed of four levels: Representation (R1), Replication (R2), Reality (R3), and Relational (R4). A DT is classified into one of the 4R phases depending on its maturity and capability levels, which increase in each phase, (see Figure 1 Digital Twin Capabilities (4R Framework). Figure reproduced from [96].Figure 1). Furthermore, each level of this framework describes the development process of a DT for a system.

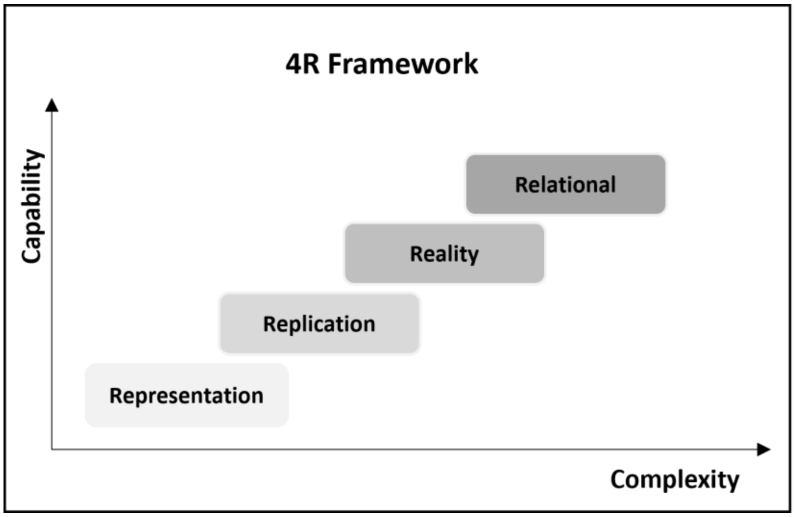


Figure 1 Digital Twin Capabilities (4R Framework). Figure reproduced from [96].

The first level, Representation, is characterized as the initial step of building a DT. It focuses on understanding the behavior of the physical system and creating a system for data collection and storage from the physical environment. In this level, the data is connected and used to describe the physical machine mathematically. This data is used for visualization, validation, and control of the virtual environment [96,97].

The second level, Replication, focuses on duplicating the system in a virtual environment utilizing the architecture created in the Representation phase. The DT is capable of reproducing the same outputs when given the same inputs from the physical system. The third level, Reality, is when the DT is used to investigate what-if scenarios with the goal of using the results obtained from the virtual runs to optimize the physical system. In this level, a model has the capability to independently provide outcomes given a set of inputs, working independently from the physical system [96,97].

The last level, Relational, describes a DT with the capabilities of incorporating decision-making technologies by using AI or ML. In this level, the physical and virtual systems are connected and the data between them is bi-directional [96,97]. Furthermore, the DT has a level of autonomy that is not found in the previous levels of capability, where it can self-adjust, identify optimal strategies, and calibrate itself. A summary of the DT capabilities is shown in Figure 2.

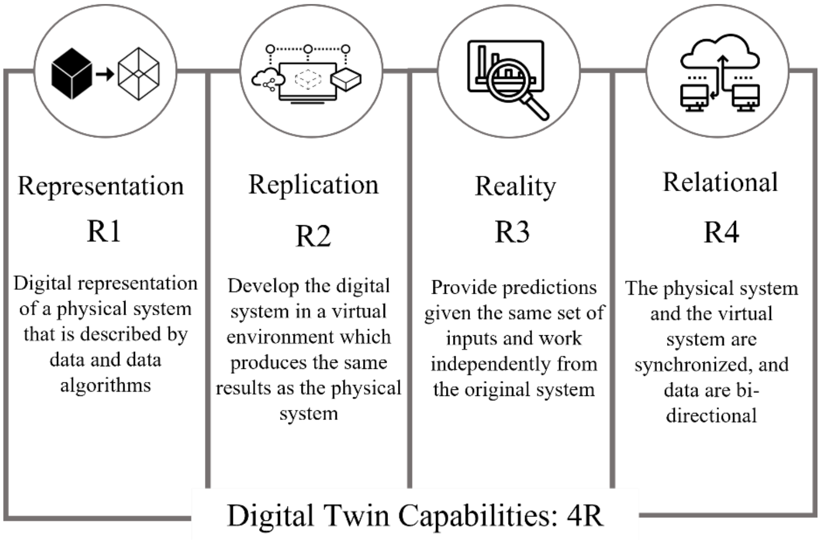


Figure 2 Summary of Digital Twin Capabilities. Figure reproduced from [96].

Overall, although the 4R framework is a general guide in defining and characterizing a DT based on the levels of capabilities, there is no current analogous framework for simulation.

In this background chapter, we have provided context of this research, exploring the concepts of Industry 4.0, VSM, simulation, and DT. This background chapter is the foundation for the following chapters and the concepts presented herein will constantly be referenced throughout this dissertation.

Chapter 3

Research Statement

3.1 Research Objective

Although traditional simulation and DT models have similar capabilities in replicating physical systems in a virtual environment, they are not interchangeable [14]. It is common to combine simulation with DT, which can cause confusion when simulation models are classified as DT and vice versa [98]. While traditional simulation is valuable in various applications, it is not accurate to refer to it as a DT. It is important to differentiate the capabilities of a DT and a simulation to achieve consistency in DT implementations. DTs and simulations are distinct technologies with unique benefits that can provide valuable insights into a problem. Therefore, it is crucial to classify them appropriately and clarify their differences to avoid misunderstandings. The main aim of this research is to clarify the distinctions between simulation and DT by exploring their respective characteristics and capabilities. The intended outcome is to provide both academia and industry with a better understanding of these two technology domains and how to bridge the gap between them, which will enable the pursuit of more effective and relevant applications and research in the future.

3.2 Research Gaps

This research addresses three gaps that are identified through the literature review:

(1) There is lack of understanding with regard to the direct comparison between the capabilities of simulation models and DT. Furthermore, existing frameworks do not clearly demonstrate how to add capabilities, sequentially, to a simulation model, specifically a DES.

(2) The use of simulation combined with a DT is very common, which creates an immense misconception about classifying a simulation model as a DT and vice versa. There is not a clear distinction between the capabilities of simulation and DT, and it is crucial to clarify their differences to reduce confusion.

(3) A systematic technique is not currently available to assist researchers and manufacturers in building a DT from a fully capable simulation.

3.3 Research Contributions

In this research, three novel contributions are proposed to fill the gaps found:

- The first contribution of this research is the development of a framework (termed 4S) that defines the levels of capability of a simulation model. An example of a successful simulation application is provided based on this framework. This approach shows how to gradually enhance the capability of a simulation model, one level at a time.
- The second contribution of this research is a systematic literature review to determine if researchers are actually using a DT approach or if they are using simulation instead of DT. The 4R and 4S frameworks are utilized to classify the works as either simulation or DT, depending on the capabilities presented.
- The third contribution of this research aims to connect simulation and DT by offering a general approach that guides researchers in building a DT directly from a fully capable simulation and implementing it in an industrial setting. By bridging the gap between the two methods, the framework allows for more efficient and effective development of DTs. This contribution highlights the potential for simulation models to serve as a foundation for the development of DTs.

Chapter 4

Contribution 1: Simulation Capabilities Framework and Application[1]

4.1 Introduction

In the creation of an approach for achieving a DT from a fully capable simulation, it is important to first understand the levels of capability of a simulation and DT individually and have a clear differentiation between the two. These two technologies have some similar capabilities and objectives, and it is important to clarify the differences to be able to bridge the gap between them. Several researchers have proposed frameworks for construction of DT, as mentioned in Section 2.5. The framework used in this research is called 4R framework [96,97]. It is the intention of this chapter to propose an analogous framework for simulation that explicitly demonstrates the levels of capability of a simulation model.

Although there is a great deal of simulation modeling cycles and frameworks defined in the literature, the levels of capabilities of simulation have not yet been addressed in a framework format that is analogous and can be directly compared to the existing 4R framework for DTs. An analogous framework for simulation capabilities should provide a similar assessment of the level of simulation maturity in an organization. This would help organizations to understand their current level of simulation capabilities, identify areas for improvement, and develop a roadmap for future simulation investments. In addition, an analogous framework would facilitate comparisons between simulation and DT capabilities, allowing organizations to better align their

---

[1] Portions of this chapter were published as [23].

simulation and DT strategies. This research aims to use the extant literature related to simulation to identify its key capabilities. To make it comparable with the 4Rs of DT, it is called the 4S framework: Modeling (S1), Analyzing (S2), Predicting (S3), and Prescribing (S4).

In the manufacturing domain, the most popular simulation technique is DES [11] and that is the method to be used to provide a case study that fulfills all the levels of capability defined in the 4S framework. DES was chosen because it can incorporate variability into the system that is difficult to capture in real life.

Specifically, to illustrate the 4S framework, a simulation of a specific process at an engine manufacturing plant was developed. A VSM was created prior to developing the DES model of the engine block machining that is demonstrated as a case study in this chapter. The main source of waste within the system was found to be in the leak-test area of this line, which causes delays, creates off-line WIP, uses excess manpower, causes waste in transportation, etc. This area was then represented in a DES model, for which a description of how to add capability and utility to the model is provided, achieving each level of the 4S framework one at a time.

This chapter is structured as follows: In section 4.2, a background related to this research is presented. Section 4.3 presents the simulation capability framework (4S). In section 4.4, a case study of the application of each level of capability of the 4S framework to an automotive engine manufacturing plant is provided. Finally, section 4.5 contains a summary of the contributions of this chapter.

4.2 Methodology: The 4S Framework

Although there are many frameworks and guidelines for building simulations, as discussed in Section 2.4, there is currently no framework that enables precisely defining the different levels of capability of a simulation model in a way that is comparable to a framework for DT. For clarification, the capabilities of simulation are well established in the literature; therefore, the goal with this framework is to aggregate the existing definitions and combine them into a framework that will serve as a guide to the development of a simulation model based on its capabilities that is analogous to the existing 4R framework for building a DT (mentioned in Section 2.5). The 4S framework for simulation should be analogous to the 4R framework for DTs to allow for classification of recent applications of DT and simulation using these two frameworks, to compare them, and to help identify and fill the gaps that exist. It is important to emphasize that there is no 1-to-1 mapping from 4S to 4R frameworks. The method employed resulted in a framework with four categories for simulation. The aim is to gather definitions of simulation from research literature that focuses specifically on simulation and its characteristics to inform a framework. To achieve this, two distinct literature searches were conducted: one for books and another for scholarly articles. The methodology for selecting the books and articles used in this research are summarized in Figure 3.

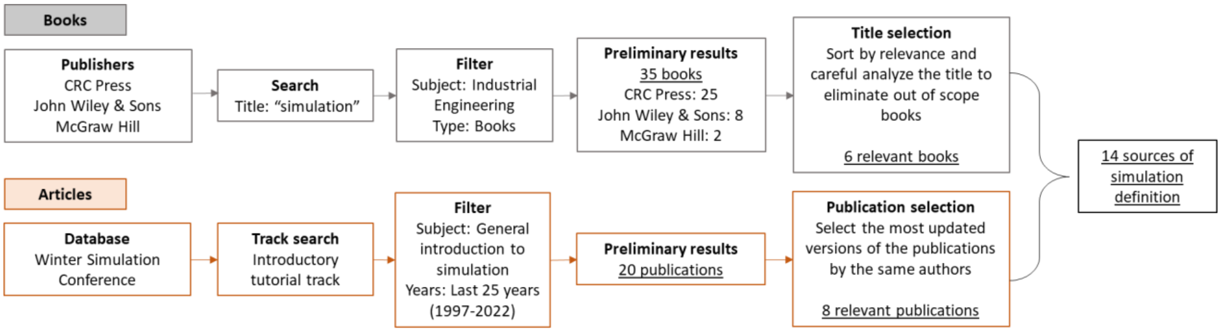


Figure 3 Systematic Search for Books and Articles.

To gather relevant literature for the book search, three major academic publishers were selected: CRC Press, John Wiley & Sons, and McGraw Hill. These publishers specialize in publishing textbooks and reference works and have a strong reputation for publishing across a wide range of academic fields. They are widely respected in the academic community and are particularly known for publishing authoritative works in the fields of engineering, technology, and computer science, which are pertinent to the study of simulation. By focusing on these publishers, the likelihood of discovering relevant books that offer comprehensive information and analysis on simulation and its features is higher.

The next step was to conduct a search to identify books that contain the term "simulation" in their title. The search was limited to books within the field of industrial engineering to ensure that the analysis remains pertinent and targeted. The books retrieved from each database, CRC Press (25 books), John Wiley & Sons (8 books), and McGraw Hill (2 books), were then combined for a total of 35 books. A detailed and rigorous examination was then undertaken to select the most appropriate books for this study. This involved sorting the results by relevance and analyzing the title of each book, selecting only those that directly provide concepts on simulation. The inclusion and exclusion criteria for this study were designed to consider only books that covered the general theory of simulation, such as handbooks, which includes its definitions and characteristics. Any books that mentioned a specific type of simulation or application field in their title, were excluded to prevent any bias towards a particular definition of simulation that might be associated with a specific simulation paradigm. These criteria ensure that the focus remains solely on the core theory of simulation and its fundamental principles, ensuring that the research is not influenced by any specific type of simulation such as Discrete-event or Multi-agent simulation. Following the initial analysis of the book titles, 19 books were found to not meet the inclusion criteria and were discarded. As a result, six unique books were identified as relevant sources for collecting definitions of simulation. These books are presented in Table 1.

To obtain relevant literature for the article search, the Winter Simulation Conference (WSC) proceedings were selected as the database. This decision was made because the WSC is a prominent conference that offers a presentation track exclusively devoted to introductory tutorials every year, which generally encompasses the topic of simulation and its characteristics. The authors of these tutorials are typically leading figures in the simulation field, and they provide thorough information on the concept and definition of simulation. This focus on definitions and concepts is not typically found in journal articles, which assume the reader already has a deep understanding of the field. A filter was applied to identify articles related to a general introduction to simulation that were published within the last 25 years in the introductory tutorial track, resulting in 20 publications. While this track offers introductory papers on various topics, the articles related to simulation are of particular interest for this research. Some of these articles were written by the same authors in consecutive years. In these cases, the articles were generally revised versions of what had been presented the previous year. Thus, the most current versions were chosen for each author, resulting in 8 relevant publications. The final set of relevant books and articles selected for this research are shown in Table 1.

Table 1 Simulation Definition Sources.

	Ref.	**Simulation Definition Sources**
Books	[37]	Law, A. M., Kelton, W. D. (2007). *Simulation modeling and analysis* (Vol. 3). New York: Mcgraw-hill.
	[8]	Banks, J. (Ed.). (1998). *Handbook of simulation: principles, methodology, advances, applications, and practice*. John Wiley & Sons.
	[99]	Chung, C. A. (2003). *Simulation modeling handbook: a practical approach*. CRC press.
	[100]	Sokolowski, J. A., & Banks, C. M. (Eds.). (2011). Principles of modeling and simulation: a multidisciplinary approach. John Wiley & Sons.
	[101]	Bossel, H. (2013). Modeling and simulation. Springer-Verlag.
	[102]	Bandyopadhyay, S., & Bhattacharya, R. (2014). Discrete and continuous simulation: theory and practice. CRC Press.
Articles	[40]	Maria, A. (1997). Introduction to modeling and simulation. In Proceedings of the 29th conference on Winter simulation (pp. 7-13).
	[103]	Shannon, R. E. (1998). Introduction to the art and science of simulation. In *1998 winter simulation conference. proceedings (cat. no. 98ch36274)* (Vol. 1, pp. 7-14). IEEE.
	[104]	Carson, J. S. (2005). Introduction to modeling and simulation. In *Proceedings of the Winter Simulation Conference, 2005.* (pp. 8-pp). IEEE.
	[47]	Banks, J. (1999). Introduction to simulation. In *Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future-Volume 1* (pp. 7-13).

	[49]	Ingalls, R. G. (2011). Introduction to simulation. In *Proceedings of the 2011 winter simulation conference (WSC)* (pp. 1374-1388). IEEE.
	[105]	Goldsman, D. (2007). Introduction to simulation. In Proceedings of 2007 Winter Simulation Conference (pp. 26-37). IEEE.
	[106]	Sanchez, P.J. (2006). As simple as possible, but no simpler: a gentle introduction to simulation modeling. In Proceedings of the 2006 winter simulation conference (pp. 2-10). IEEE.
	[107]	White, K. P., & Ingalls, R. G. (2018). The basics of simulation. In 2020 Winter Simulation Conference (WSC) (pp. 147-161). IEEE.

The next step involved collecting and analyzing the definitions and features of simulation presented in the selected books and articles to identify the main characteristics. The definitions were transcribed verbatim to an excel file with the intention of organizing them into a unique framework that is focused on the levels of capability of a simulation. A thorough process of transcribing, analyzing, and categorizing the simulation definitions gathered was conducted. To accomplish this, each clause within each definition was separated into individual cells in a spreadsheet and analyzed by identifying the action verb of each clause. By breaking down the definitions into their component parts and grouping them based on common themes, six main characteristics of simulation were identified and labeled as C1 through C6. For instance, many of the phrases used the action verb "understand", which suggests that these phrases were focused on understanding a particular system or process. By grouping together phrases with similar action verbs, a characteristic of "Understanding how the system works" was identified and labeled as C1. Similarly, the action verbs "build" and "create" were commonly used in many clauses, indicating that these phrases were focused on the capability of simulation to create a model of a system. These phrases were categorized as C2, which is named “Represent the physical system in a digital model”. A similar process was followed to arrive at C3 through C6.

Table 2, Table 3, Table 4, Table 5, Table 6, and Table 7 contain the main characteristics of simulation identified in the collected literature. A summary of these characteristics is shown in Table 8.

Table 2 Simulation Characteristics (C1).

(C1) Understand how the system works	**Reference**
Gain some understanding of how the corresponding system behaves	[37]
Aid in providing understanding about how a system really operates	[8]
Understand the complexity of the system	[100]
Understand the real system	[101]
Understand the system	[102]
Understand the behavior of the system	[103]
Understand the behavior of the system	[104]
Understand the system behavior	[47]

Table 3 Simulation Characteristics (C2).

(C2) Represent the physical system in a digital model	Reference
Build a model as a representation of the system	1
Imitation of the operation of a real-world process or a system over time	2
Create a computerized mathematical model of a physical system	3
Imitate the real world process or system over time	4
Representation of a system	4
Model the actual process of a system	5
Represent the behavior of a system in a computer	5
Imitation of a system through a model	6
Create a model that is a close approximation to the real system and incorporate most of its salient features	7
Construction of a model	8
The model mimics the response behavior of the real system to events that take place over time	8
Design a model of a real system	8
Design a model of a real system	9
Mimic the dynamics of a real system	9
Representation that incorporates time and the changes that occur over time	10
Imitation of the operation of a real-world process or system over time	11
Imitation of the operation of a real-world system	12
Computer model that mimics describe the behavior of the system being modeled	13
Model imitates some salient aspect of the behavior of the system under study	14
Create a model which imitates the behaviors of interest	14

Table 4 Simulation Characteristics (C3)

(C3) Examine and analyze the system and its components	**Reference**
Study the model as a surrogate for the actual system	[37]
Describe and analyze the behavior of a system	[8]
Analyze systems	[99]
Understand the complexity of the system	[100]
Simulate the behavior of a system	[101]
Design experiments and analyze results	[102]
Conduct experiments with the model	[103]
Conduct experiments with the model	[104]
Analyze results	[104]
Describe and analyze the behavior of a system	[49]
Analyze systems	[105]

Table 5 Simulation Characteristics (C4).

(C4) Draw inferences about the system	**Reference**
Simulate the operation of the system as it currently exists	[37]
Numerically exercise the model for the inputs in question to see how they affect the output measures of performance	[37]
Draw inferences concerning the operating characteristics of the real system that is represented	[8]
Gain insight into the operation of a system	[99]
Draw conclusions about the system	[100]
Infer properties concerning the behavior of the actual system or its subsystem	[40]
Gain insights into how a modeled system actually works and understanding of which variables are most important to performance	[103]
Provide insight into system performance	[47]
Draw inferences concerning the operating characteristics of the real system that is represented	[49]
Gain insights	[106]
Experiment with the model to infer the system behavior	[107]

Table 6 Simulation Characteristics (C5).

(C5) Evaluate and predict how the physical system will behave under different conditions	**Reference**
Predict performance under some new conditions being considered	[37]
Simulate the operation of the system as it would be if the system were expanded	[37]
Use a computer to evaluate a model numerically, and gather data to estimate the desired true characteristics of the model	[37]
Ask what-if questions about the real system	[8]
Test new concepts and/or systems before implementation	[99]

Gain information without disturbing the actual system	[99]
Test hypothesis about the system	[100]
Predict potential outcomes	[100]
Predict how a system will respond under certain conditions	[101]
Observe how the system behaves under different conditions	[101]
Test hypothesis	[101]
Ask what-if questions	[102]
Predict the effect of changes to the system	[40]
Answer what if questions	[40]
Evaluate the performance of a system, existing or proposed, under different configurations of interest	[40]
Experiment with new and unfamiliar situations and to answer what-if questions	[103]
Test hypothesis	[103]
Test new designs, layouts, etc. without committing resources to their implementation	[103]
Predict future behavior	[103]
Evaluate various strategies for the operation of a system	[104]
Predict the effect of proposed changes	[47]
Experimentation with a model of a system	[47]
Evaluate comparison of many alternative designs	[47]
Ask what-if questions about the real system	[49]
Analyze and compare certain scenarios quickly and efficiently	[105]
Study how a prospective system will work	[106]
Test and compare alternative designs	[107]
Evaluate different scenarios	[107]

Table 7 Simulation Characteristics (C6).

(C6) Provide insight into the system's optimal operation to support decision making	**Reference**
Study the systems to try to gain insight into optimal relationships among various components	[37]
Aid in the design of real systems	[8]
Make operating or resource policy decisions	[99]
Make recommendations based on the interactions of the simulation	[100]
Provide insights to decision making	[100]
Prescribe optimal solutions	[100]
Provide insights to actions that need to be made to the system	[101]
Able to support decision making	[102]
Optimize system performance	[40]
Evaluating various strategies for the operation of the system	[103]
Decision Making	[103]
Recommend alternatives	[47]
Problem-solving methodology for the solution of many real-world problems	[49]

Allows for methods for selecting the best of a number of competing scenarios	[105]
Address a specific set of questions	[106]
Study recommendations	[107]

Table 8 Summary of the Simulation Characteristics.

Characteristics (Cn)
(C1) Understand how the system works
(C2) Represent the physical system in a digital model
(C3) Examine and analyze the system and its components
(C4) Draw inferences about the system
(C5) Evaluate and predict how the physical system will behave under different conditions
(C6) Provide insight into the system's optimal operation to support decision making

The subsequent action involved referring back to the books and articles to identify which characteristics (C1-C6) were explicitly cited in their definitions of simulation. Table 9 shows a summary of this classification process, indicating that while some features are implied, some authors do not explicitly state them in their definitions, as observed for C1, C3, and C4. On the other hand, all sources of definitions explicitly mention that simulation is a digital representation of a system (C2), it is used to evaluate and predict how the physical systems behave under different conditions (C5), and to provide insight for decision making (C6). Hence, after examining the individual definitions of each characteristic (C1-C6), it was found that certain categories had significant similarities in their concept. Consequently, C1 and C2 were merged into a single category, and C3 and C4 were also combined.

The categories (C1) "Understand how the system works" and (C2) "Represent the physical system in a digital model" were combined into a single category because many authors combine understanding and representation of a system into one step [8,37,47,100–104]. C1 emphasizes the need to understand the behavior of a system to be able create an accurate simulation model. This involves identifying the various components of the system, how they interact, and their relationships [108]. On the other hand, C2 involves the creation of a digital representation of a physical system. This involves creating a simulation model that accurately reflects the characteristics and behavior of the physical system. To do this, the simulation modeler needs to have a deep understanding of how the physical system works, its components, and how they interact [35].

The categories (C3) "Examine and analyze the system and its components" and (C4) "Draw inferences about the system" were combined into a single category because they both involve the process of analyzing a system to draw conclusions about its behavior. C3 involves examining and analyzing the various components of a system in the simulation model to understand how they interact with one another. This analysis can include identifying the relationships between different components and how they affect the behavior of the system as a whole. Similarly, C4 is concerned with drawing inferences about the behavior of the system based on the results of the simulation model. Once the simulation is complete, the modeler can analyze the results and draw conclusions about the system's behavior [39].

After combining similar characteristics, the result was a framework consisting of four categories. To make it more comparable it with the 4R framework of DT, these categories are defined as the 4S framework of simulation: S1 (combining C1 and C2), S2 (combining C3 and C4), S3 (C5), and S4 (C6).

To appropriately label the four categories in the 4S framework of simulation, the most common and relevant terms associated with each category were visually represented in a word cloud. This method helped to simplify and condense the keywords, making it easier to identify the most important and frequently used terms. By analyzing the word cloud, patterns and trends can be identified, which can aid in defining or summarizing the key features of simulation. The word clouds for each category are shown in Figure 4.

Table 9 Simulation Definition Sources Classification.

	Ref.	**Simulation Definition Sources**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
Books	[37]	Law, A. M., Kelton, W. D. (2007).	X	X	X	X	X	X
	[8]	Banks, J. (Ed.). (1998)	X	X	X	X	X	X
	[99]	Chung, C. A. (2003).		X	X	X	X	X
	[100]	Sokolowski, J. A., & Banks, C. M.(2011).	X	X	X	X	X	X
	[101]	Bossel, H. (2013).	X	X	X		X	X
	[102]	Bandyopadhyay, S., & Bhattacharya, R. (2014).	X	X	X		X	X
Articles	[40]	Maria, A. (1997).		X		X	X	X
	[103]	Shannon, R. E. (1998).	X	X	X	X	X	X
	[104]	Carson, J. S. (2005).	X	X	X		X	X
	[47]	Banks, J. (1999).	X	X		X	X	X
	[49]	Ingalls, R. G. (2011).		X	X	X	X	X
	[105]	Goldsman, D. (2007).		X	X		X	X
	[106]	Sanchez, P.J. (2006).		X		X	X	X
	[107]	White, K. P., & Ingalls, R. G. (2018).		X		X	X	X
			S1		S2		S3	S4

After analyzing the word clouds, names were assigned to each of the four categories in the 4S framework of simulation. It was decided that for each category to represent a capability, action verbs would be favored when choosing the category labels. In the word cloud of the first category (S1), the most frequently used terms were “model”, “behavior”, “understand”, and “real”. The words “model” and “behavior” suggest a focus on simulating the behavior of a system. Similarly, the term “understand” and “real” indicate a focus on comprehending the workings of the real system. These words are strongly associated with conceptual and computer models, suggesting that "Modeling" is an appropriate name for this category. The largest and most frequently occurring words in the word cloud for second category (S2) were "analyze" and "behavior". These terms suggest that this category focuses on the examination and evaluation of system behavior and performance, making "Analyzing" an appropriate name for this category. In the word cloud for the third category (S3), the most frequently occurring words were “predict”, “hypothesis”, “what-if”, “test”, and “evaluate”. These terms suggest that this category focuses on the use of simulation to predict how a system will behave under different conditions and to test and evaluate hypothetical scenarios. Therefore, “Predicting” is an appropriate name for this category. Finally, in the word cloud of the fourth category (S4), the most frequent words were “decision”, “making”, “provide”, “insights”, and “recommendations”. These terms suggest that this category focuses on the use of simulation to provide valuable insights and recommendations that can inform decision making. The term “prescriptive” is commonly used in the field of computing and analytics to refer to a type of analytics that involves recommending specific actions to achieve a desired outcome [109]. Therefore, “Prescribing” is an appropriate name for this category.

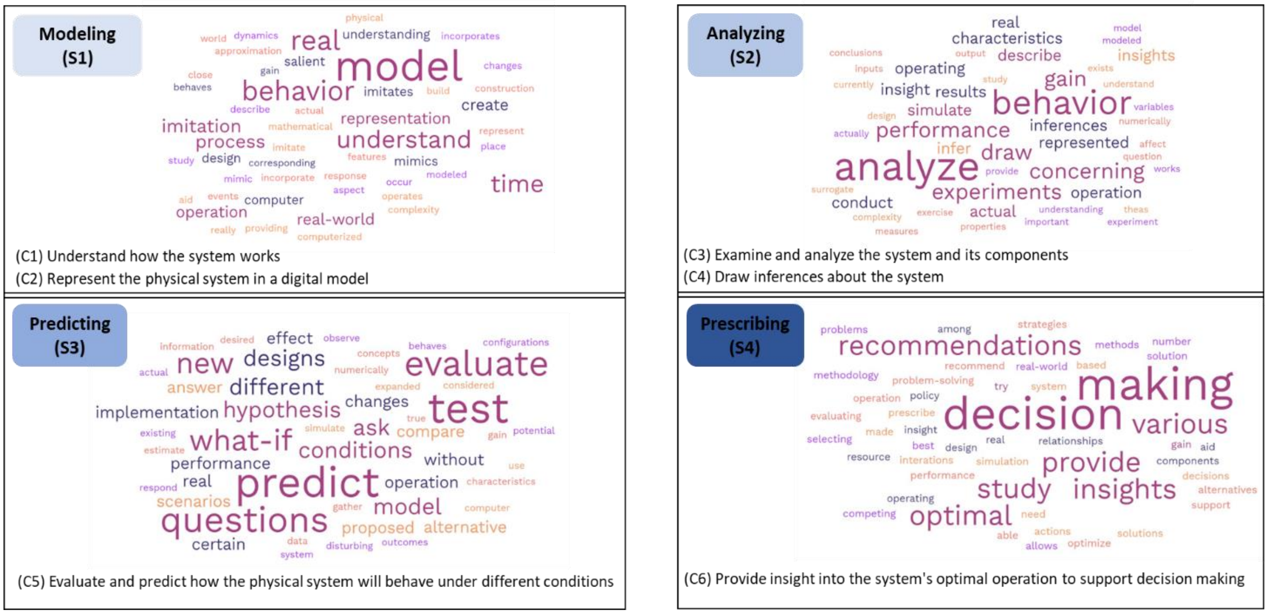


Figure 4 Word Clouds for Categorization of the 4S Levels.

Overall, the 14 authors shown in Table 1 seem to agree that a simulation is capable of modeling and representing the physical system in a virtual environment (S1), providing analysis of the system (S2), predicting what would happen when applying changes to the system (S3), and prescribing optimal actions for the system (S4).

This 4S framework is an approach to the development of a simulation model, each level building upon the previous level providing increased utility and capability. This framework is shown in Figure 5 and the various levels of the 4S framework are described in detail in the following sections. The x-axis in Figure 5 represents the analytical capability of a simulation, which refers to the ability to analyze information or data of complex systems. The y-axis represents the analytical utility of a simulation, which refers to the usefulness of a simulation model, such as offering insights, analysis, and decision support. As the levels of capabilities of a simulation model increase, the level of utility of the model also increases [110]. This is because a more capable simulation model is able to provide more accurate and detailed insights into the real-world phenomena being simulated. A simulation model with greater capabilities is able to incorporate more detailed and accurate data into the model, resulting in more precise and reliable predictions [37]. It is also more flexible and can be used to simulate a wider range of scenarios and variables. Overall, the level of utility of a simulation model increases as its capabilities increase, allowing users to make more informed decisions and gain deeper insights into complex systems and processes [100]. A summary of these levels of capability is shown in Figure 6.

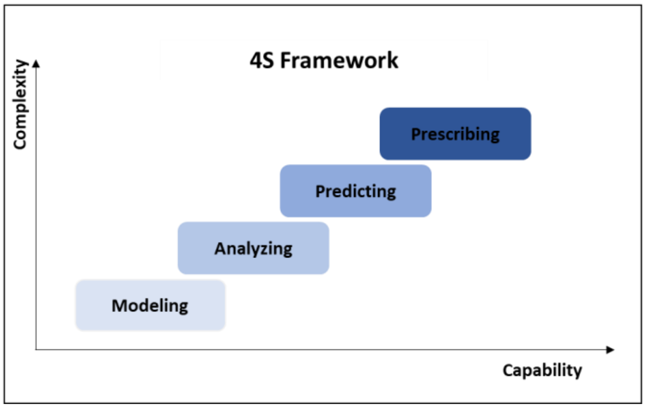


Figure 5 Simulation Capabilities (4S Framework).

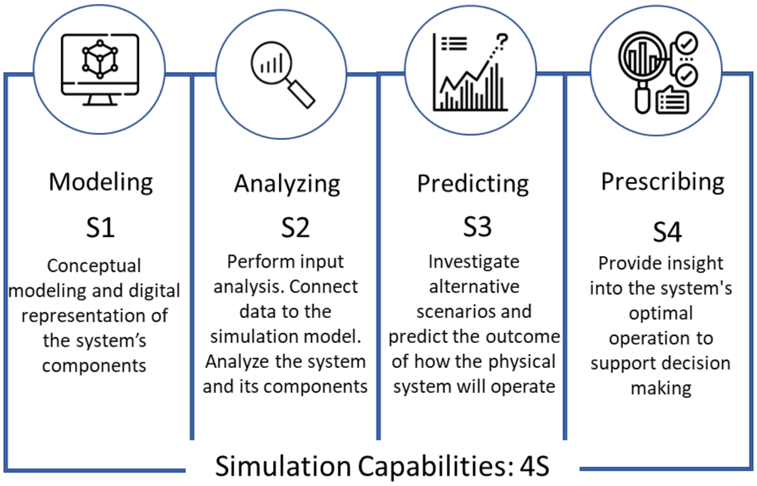


Figure 6 Summary of Simulation Capabilities (4S).

4.2.1 Modeling (S1)

The first level of simulation capability, Modeling, comprises two parts - conceptual modeling and computer modeling. During the conceptual modeling stage, the modeler creates a simplified representation of a real-world system or process by using concepts, relationships, and assumptions that capture the essential features of the system [108]. The purpose of conceptual modeling is to gain a thorough understanding of the system, to map out the process, collect the requirements, understand the constraints, and build the architecture of the system to be modeled

[36]. Skipping the conceptual stage and moving straight to building a model could lead to a non-validated and non-verifiable model that fails to capture the process and produces inaccurate simulation results [54].

In the computer modeling stage, the system's components are represented in a digital format, and a simulation model is created [24]. The modeler collects the required data for the simulation and comprehends it before using it as input for the model [35]. The modeler may use various tools, such as VSM or data mapping, to gain a better understanding of the system during this phase [21]. Although a digital representation of the system is created, this phase is mostly conceptual, and no data is connected, nor is any analysis performed.

To achieve the Modeling level, the modeler must first understand the system and its behavior, by identifying its key properties, characteristics, and states. The next step is to collect data and determine the measures of performance. This involves gathering data on the system's behavior and performance, and defining the metrics that will be used to evaluate the model's accuracy and effectiveness [39]. After collecting data and defining performance measures, the modeler can begin to represent the components of the physical system in a computer model [100]. This involves selecting an appropriate modeling technique and creating a computational model that accurately represents the behavior of the system [54]. The modeler must also select appropriate parameters, boundary conditions, and initial conditions for the model, based on their understanding of the physical system and the data collected [47].

In the Modeling level, important decisions regarding the purpose of the simulation should be made. Once these decisions are made, the Modeling level is considered complete [8]. It is important to note that although a digital model is created during this stage, data connection, model verification and validation, and system analysis are carried out in the next level, Analyzing (S2).

The completion of the Modeling level provides a solid foundation for the modeler to proceed to the next level of capability: Analyzing.

4.2.2 Analyzing (S2)

After representing the system in the virtual space at the Modeling level, the model can be analyzed. In the Analyzing level, the modeler incorporates the inputs and information gathered about the physical system in the Modeling level (S1) into the model. In this phase, the modeler can conduct input analysis, connect the data collected to the model, verify and validate the model, and perform sensitivity analysis on the system in its current state to draw inferences and analysis about the existing system that is being modeled, from observations in the simulation model.

To achieve the Analyzing (S2) level of the 4S framework, the modeler performs an input analysis to determine the data that will be used as input to the simulation model and to ensure that it adequately approximates the real-world behavior of the system being modeled. There are several ways to represent the data that is collected in this phase in a simulation model [55]. If the variable is independent of other variables, it can be assumed that it is deterministic, fit a probability distribution to the data, or use the empirical distribution of the data [39]. When assuming that the data is deterministic, or constant, it can be obtained by averaging historical data. However, this technique can invalidate results if there is randomness in the model [37]. On the other hand, when there are sufficient data points, fitting probability distribution to the data may be more appropriate. The empirical distribution can be used to describe a sample of observations of a given variable [37]. Once the input analysis is performed, the modeler can perform verification and validation of the model to ensure the accuracy and reliability of a model. Verification refers to the process of determining the correctness of the model implementation, while validation refers to the process of ensuring whether the model accurately represents the real-world system it is intended to simulate [111].

After the model is verified and validated, the modeler can develop a design of experiments that will provide sensitivity analysis of the system under current and future parameter values. Once the simulation can provide analysis and inferences of the physical system in its current state, it has successfully reached the Analyzing level of the 4S framework, which allows the modeler to add the next level of capability: Predicting.

4.2.3 Predicting (S3)

The Predicting level is where the simulation has the capability to predict how the real system will operate under new and different conditions. It can be used to investigate and evaluate alternative scenarios and analyze results from varying the model inputs and structure. It also assists in tracking statistics to measure and compare the performance of a system. In this stage, it is important to consider developing accurate and realistic alternative scenarios that can potentially be applied in the real system [36,39,41].

The main characteristics of the Predicting level are the ability to calculate how the system would behave when evaluating the results from the run of the alternative scenarios. Here, the results of the experiments are compared within scenarios, as well as across scenarios, relative to the current state of the system. This level involves examining the real system, its components, and their relations to understand the potential outcomes and future states under a range of uncertain factors with the objective of assessing performance of the physical system or one of its major elements.

To add this level of capability to a simulation, the modeler defines alternative scenarios, evaluates the results from the run of these alternative scenarios, predicting how the changes made in the model would impact the measurements of performance of interest. Since the Computer model is verified and validated, trusting that it is accurately representing the system the results from the alternative scenarios can predict outcomes and how the system would behave [111].

4.2.4 Prescribing (S4)

The last level, Prescribing, is achieved when a simulation model is used to evaluate the system and provide insight into its optimal operation or configuration to support decisions. The Prescribing capability is an advanced feature of simulation modeling that can be particularly useful for decision-making purposes. It can help decision-makers identify the most effective course of action to achieve their goals and provide insights into the potential outcomes of various intervention strategies.

In this level, decision-support activities, or even decision-making, are performed, which can be done using simulation optimization and/or aided by ML. The simulation optimization approach is when optimization techniques are integrated into simulation modeling and analysis, this can include any approach that iteratively and systematically adjusts parameters and uses the simulation to evaluate each alternative until it reaches some stopping condition [8]. The second approach, aided by ML, is when machine learning methods, such as neural networks are used to leverage big data and improve performance either within the simulation or as part of a larger ML algorithm of which the simulation is just one part [112].

4.3 Case Study: The Machining Line in an Automotive Engine Manufacturing Plant

The automotive engine block manufacturing plant studied for this research is segregated into three sections: die-casting, machining, and engine assembly. The scope of this study is limited to the engine block machining line. Figure 7 presents a high-level VSM of the engine block machining line, which is comprised of 20 stations, represented in grey boxes, that perform different machining operations on engine blocks. Consecutive stations are connected by conveyors, which are represented in orange boxes. The current state VSM was developed by following the path of an engine block backwards from customer (Engine Assembly) to supplier (Block Die-Casting) and drawing a visual representation of every process in the product flow. This engine block machining line includes a few parallel and sequential sub-processes. These processes were taken into consideration while building the VSM but are not depicted in detail in Figure 7 to provide anonymity for our industry partner.

The current state VSM is based on the current practices at the facility and its information and data were gathered from two methods. First, a time study was performed to collect the cycle time and lead time of each station on the engine block machining line. Second, data such as down time was acquired by directly soliciting input from employees. Production requirements, cycle time, lead time, change over time, number of operators, available time, and inventory numbers were some of the main data collected to build the current state VSM. Detailed information from the original VSM, including specific steps and duration for each process, as well as value-added and non-value-added time, has been redacted to protect the automotive company's intellectual property. The current state VSM was constructed to identify existing sources of waste within the system, and it is represented in Figure 7.

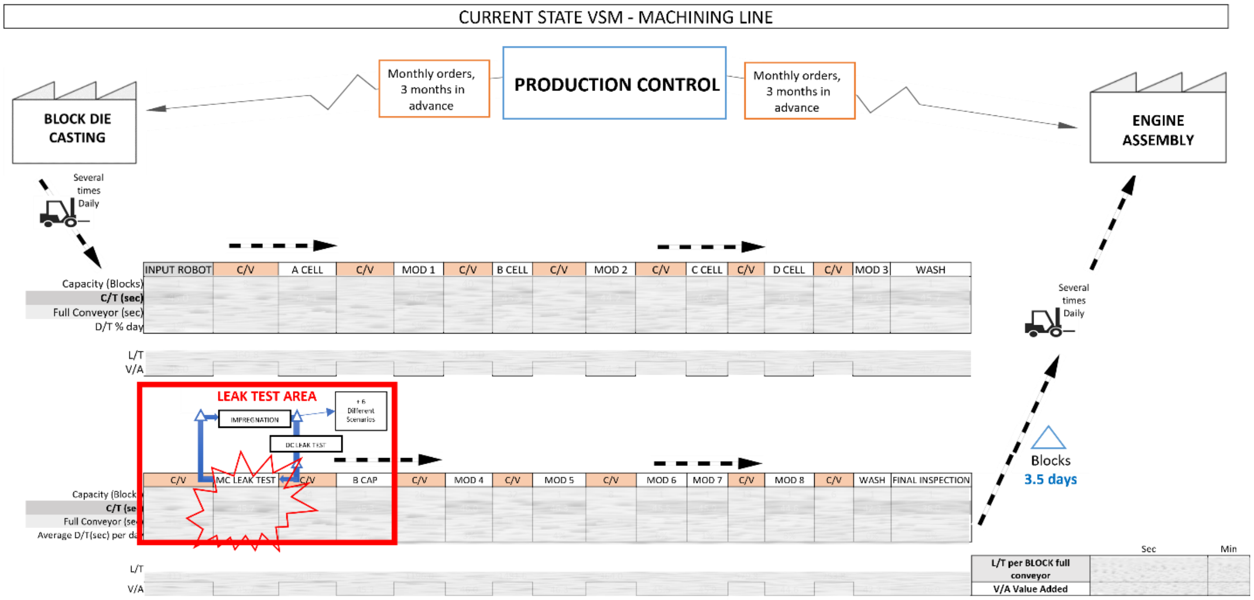


Figure 7 Current State Value Stream Map.

After creating and analyzing the current state VSM, the main source of waste was found to be in the Leak Test area, which is represented inside the red rectangle in Figure 7. The Leak Test area causes delays, off-line Work in Process (WIP), extra processing in the form of impregnation and re-testing, and excess transportation. Additionally, because the Leak Test area has a complex and stochastic routing structure, it was not possible to calculate a lead time for the leak test process when building the VSM. For these reasons, it was decided to develop a fully capable simulation model of the current state of this area, to estimate the average lead time that a block spends in this operation, and to study different scenarios/approaches that could reduce the waste generated here.

As outlined in section 2.4, DES is the predominant simulation type utilized in the manufacturing industry due to its ability to simulate discrete events that occur at specific time points [39]. The modeler can use it to observe particular events that are result from the operational procedures, and it has the potential to be applied in analyzing various kinds of systems and yielding a diverse range of outcomes [37]. Therefore, DES was selected as the preferred simulation for the case study in the automotive manufacturing company studied. Among the various software options available for simulating discrete events, Simio was selected as the preferred simulation software for this research due to its wide-ranging features that allow for the validation and execution of complex systems. Its object-oriented architecture streamlines the modeling process, and its 3D visualization capabilities make it easier for users to comprehend and communicate simulation findings. In addition, Simio includes tools for optimization that can help users identify optimal solutions for complex problems, which can be particularly valuable for systems with multiple variables and constraints, such as the Leak Test area.

4.3.1 Applying Modeling (S1)

As mentioned in Section 4.2, the first level of capability and utility of a simulation model is Modeling (S1). To develop a modeling capability in a simulation, it is necessary to first understand the physical system and identify its key properties, characteristics, and states. This understanding is used to develop a conceptual model, which is a high-level representation of the system that captures its essential behavior and interactions [37]. As shown in Figure 7, the VSM was constructed to help the modeler identify areas of waste, bottlenecks, and inefficiencies in the system, and develop strategies for improving its performance. This information was used to develop a conceptual model of the system, which can be used to construct a simulation model and conduct experiments to explore the behavior of the system under different conditions. In the area being investigated there are two leak test machines: Machining Leak Test (MC Leak Test) and Die Cast Leak Test (DC Leak Test). Machining Leak Test is considered the Master Leak Test and it is an automated process where the machine makes the decision on the condition of the block. Die Cast Leak Test is operated manually. The operator gets the results from the machine and decides on the condition of the block. Both machines test leaks and are programmed to monitor its performance to ensure that the engine block meets the required standards. The majority of the blocks that go through MC Leak Test pass the test, so they stay on the conveyor and move to the next process, which is called B-CAP. The blocks that fail MC Leak Test can be classified as large leaker, pull-off, or impregnation. A large leaker block is identified when MC Leak Test determines that the leak exceeds the required quality standards of the area, so this block is removed from the conveyor and put on a skid to be retested in DC Leak Test. A pull-off block is a block that is pulled-off the line because the conveyor that goes to impregnation is full or in danger of blocking (and hence shutting down) MC Leak Test. An impregnation block is an engine block that needs to go through the impregnation process to be repaired. All blocks that fail MC Leak Test are eventually tested in DC Leak Test, whether they are pull-offs, impregnation, or large leaker. DC Leak Test determines if the block is to be scrapped, if it needs another round of impregnation, or if it passes the test. A block that passes DC Leak Test is routed to be tested again in MC Leak Test, as that is considered the Master Leak Test.

The necessary assumptions to develop the simulation model were then defined. First, those stations that were outside the scope of the model were assumed to always be functioning, thus never blocking or starving the stations that were modeled. Therefore, the suggested improvements would be fully effective for the overall factory's production process only if the assumption holds true. Second, lunch breaks, scheduled quality checks, and any other scheduled breaks were embedded in different work schedules for each station. There are two operators in the Leak Test area that are in charge of off-loading blocks from the line (and into a WIP staging area) if the conveyors are full and putting WIP blocks back onto the line if the conveyors are empty.

Historical data was collected to be fitted into probability distributions for each machine. Downtime, processing time, availability, shifts, flows, and reliability logic were gathered from the data used to create the VSM. Building the VSM before embarking on building a DES not only helped to identify the critical process to simulate but it also allowed the modeler to become familiar with the system and facilitated the data collection [22]. A simulation of this area will allow the engine blocks to be tracked and many measures of performance will be captured.

After collecting the data, the metrics used to analyze the system in further phases are determined. In this case study, the Baseline model tracks five measures of performance:

a. WIP ready for B-CAP (WIP B-CAP): Off-line WIP waiting to go through the B-CAP process.
b. WIP ready for MC Leak Test (WIP MCLT): Off-line WIP waiting to go through MC Leak Test Machine.
c. Line Throughput: The total number of blocks that go through B-CAP and leave the system.
d. Impregnation rates: Number of blocks and how many times that each goes through impregnation.
e. Scrap rates: How many blocks are discarded as scrap.

The current state of the Leak Test area was conceptually modeled as “Baseline Model” (BL) and it is represented in Figure 8. In this model, engine blocks arrive at the MC Leak Test machine and depart going through the B-CAP process. Since MC Leak Test is the Master Leak Test, every block that goes through DC Leak Test needs to go through MC Leak Test again.

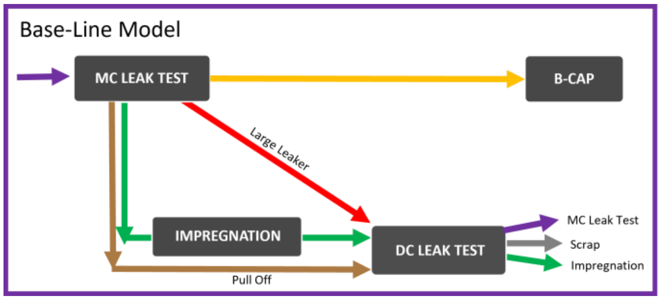


Figure 8 Leak Test Area in Baseline Model.

Once the current state of the system was understood, it was possible to represent the components of the conceptual model in a virtual environment. Figure 9 shows a snapshot of the simulation model using Simio.

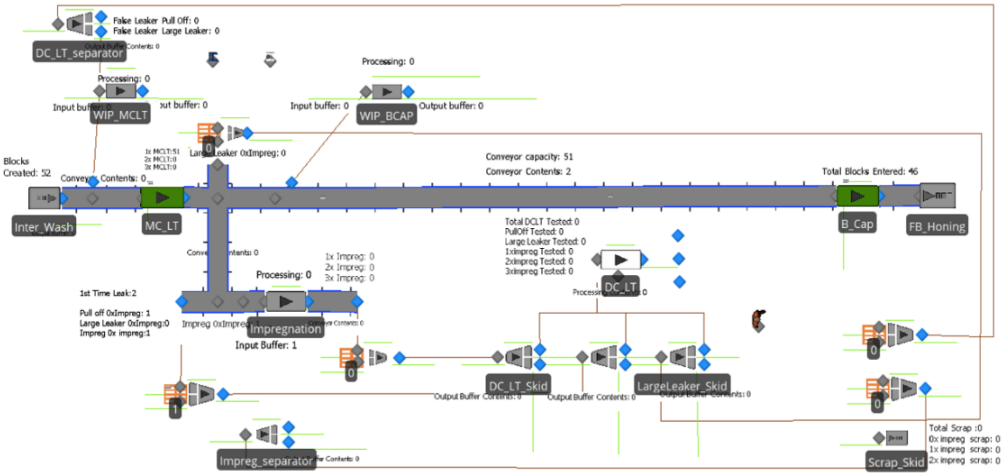


Figure 9 Snapshot of the Simulation Model.

The Modeling (S1) level of the 4S framework was achieved by understanding the system and collecting all the necessary information to represent the components of this system in a computer model. Once this is done, the next level of capability, Analyzing (S2), could be added.

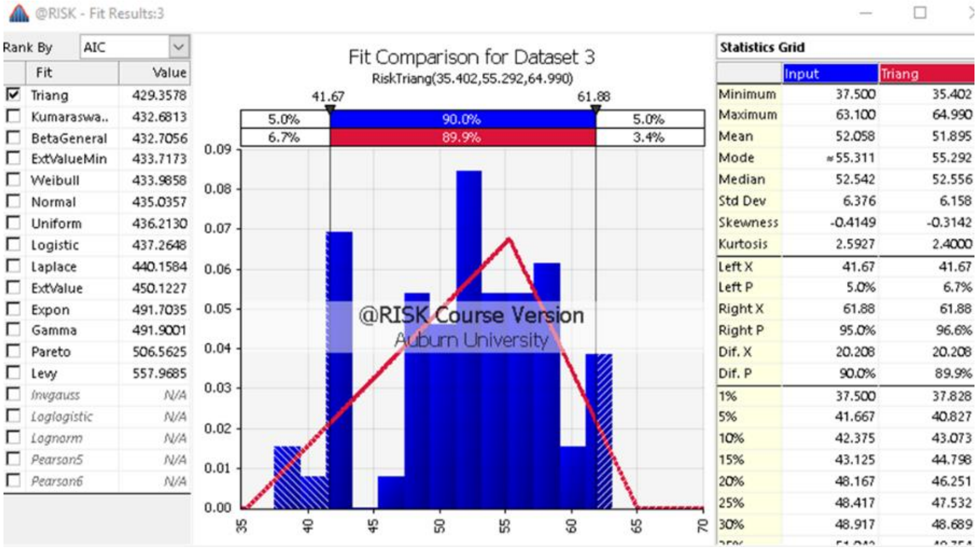
4.3.2 Applying Analyzing (S2)

In the second level of the 4S framework, Analyzing, the modeler is able perform input analysis and connect the historical data that was collected in the previous phase to the computer model, verify and validate the model, and perform sensitivity analysis on the system in its current state.

4.3.2.1 Input Analysis

In this case study, the historical data collected were fitted into probability distributions for each machine and fed into the model elements, which were used to define the arrival time of engine blocks in the model and to define the processing time and reliability logic of each machine. In this case study, the primary input analysis involved determining the arrival rate, as well as assessing the reliability logic of the MC Leak Test and the B-CAP. Other data utilized in the system were collected during the VSM analysis and subsequently utilized in the simulation model.

To determine the arrival time rate input for the simulation model source data was gathered over a period of three months. This data was then analyzed using @Risk to identify the most suitable distribution that accurately represents the collected data. The arrival time data was fitted with various distributions, and based on the use of Akaike information criterion (AIC), a triangular distribution was found to be the best fit. This distribution was selected and used as the input for the source as the arrival rate. Figure 10 displays this distribution.



Rank By: AIC

	Fit	Value
☑	Triang	429.3578
☐	Kumaraswa..	432.6813
☐	BetaGeneral	432.7056
☐	ExtValueMin	433.7173
☐	Weibull	433.9858
☐	Normal	435.0357
☐	Uniform	436.2130
☐	Logistic	437.2648
☐	Laplace	440.1584
☐	ExtValue	450.1227
☐	Expon	491.7035
☐	Gamma	491.9001
☐	Pareto	506.5625
☐	Levy	557.9685
☐	*Invgauss*	*N/A*
☐	*Loglogistic*	*N/A*
☐	*Lognorm*	*N/A*
☐	*Pearson5*	*N/A*
☐	*Pearson6*	*N/A*

Statistics Grid

	Input	Triang
Minimum	37.500	35.402
Maximum	63.100	64.990
Mean	52.058	51.895
Mode	≈55.311	55.292
Median	52.542	52.556
Std Dev	6.376	6.158
Skewness	-0.4149	-0.3142
Kurtosis	2.5927	2.4000
Left X	41.67	41.67
Left P	5.0%	6.7%
Right X	61.88	61.88
Right P	95.0%	96.6%
Dif. X	20.208	20.208
Dif. P	90.0%	89.9%
1%	37.500	37.828
5%	41.667	40.827
10%	42.375	43.073
15%	43.125	44.798
20%	48.167	46.251
25%	48.417	47.532
30%	48.917	48.689

Figure 10 Fitting Distribution to Arrival Time of Blocks in the System.

To assess the reliability of the MC Leak Test and B-CAP servers, downtime data was gathered for these machines and entered into @Risk to create a distribution for the time between failures and the time to repair. Uptime between failures refers to the duration of time during which the machine remains operational without experiencing any failures or breakdowns. Time to repair, on the other hand, refers to the amount of time it takes to fix a machine after it has failed or experienced a breakdown. Figure 11 displays the distributions used for the MC Leak Test Downtime. A uniform distribution was selected to represent the uptime between failures and an exponential distribution was selected to represent the time to repair. These distributions were selected by using the AIC criterion and visual assessment and they were incorporated into the model as shown in Figure 12.

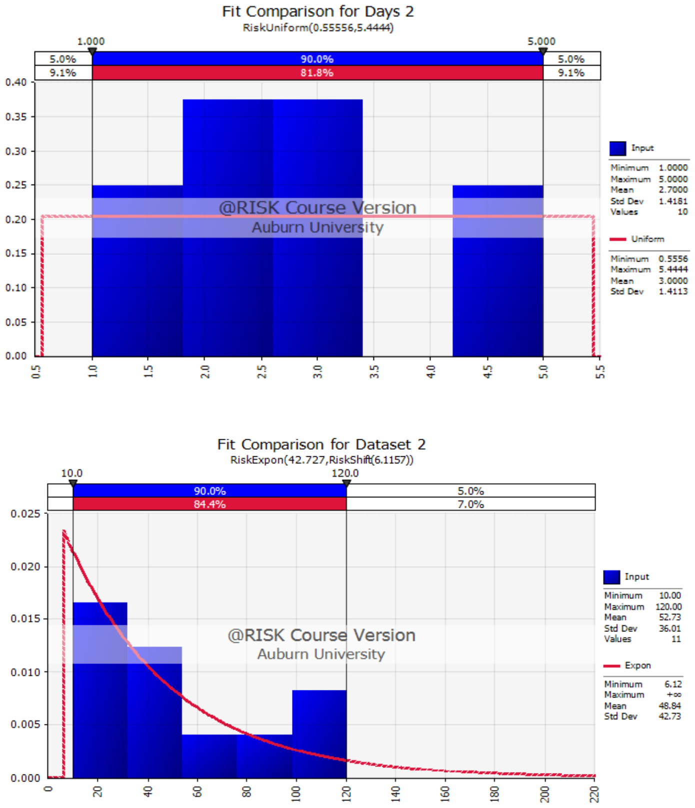
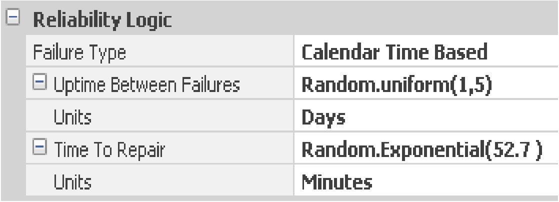


Figure 11 Fitting Distribution to MC Leak Test Downtime.



⊟ Reliability Logic	
Failure Type	Calendar Time Based
⊟ Uptime Between Failures	Random.uniform(1,5)
Units	Days
⊟ Time To Repair	Random.Exponential(52.7)
Units	Minutes

Figure 12 Reliability Logic for MC Leak Test Machine.

The same process was performed for obtaining the reliability logic for the B-CAP machine. The downtime was fitted into a distribution to find the uptime between failures and time to repair, which are shown in Figure 13.

⊟ Reliability Logic	
Failure Type	Calendar Time Based
⊟ Uptime Between Failures	Random.exponential(1.35)
Units	Days
⊟ Time To Repair	Random.Triangular(10,105.13,270)
Units	Minutes

Figure 13 Reliability Logic for B-CAP Machine.

Finally, to identify the pathways that an engine block might follow within the system and their corresponding percentages, an extensive analysis was conducted on three months' worth of data. The results of this analysis are presented in Figure 14. The next step is to perform verification and validation to be able to trust that the model accurately represents the system being modeled.

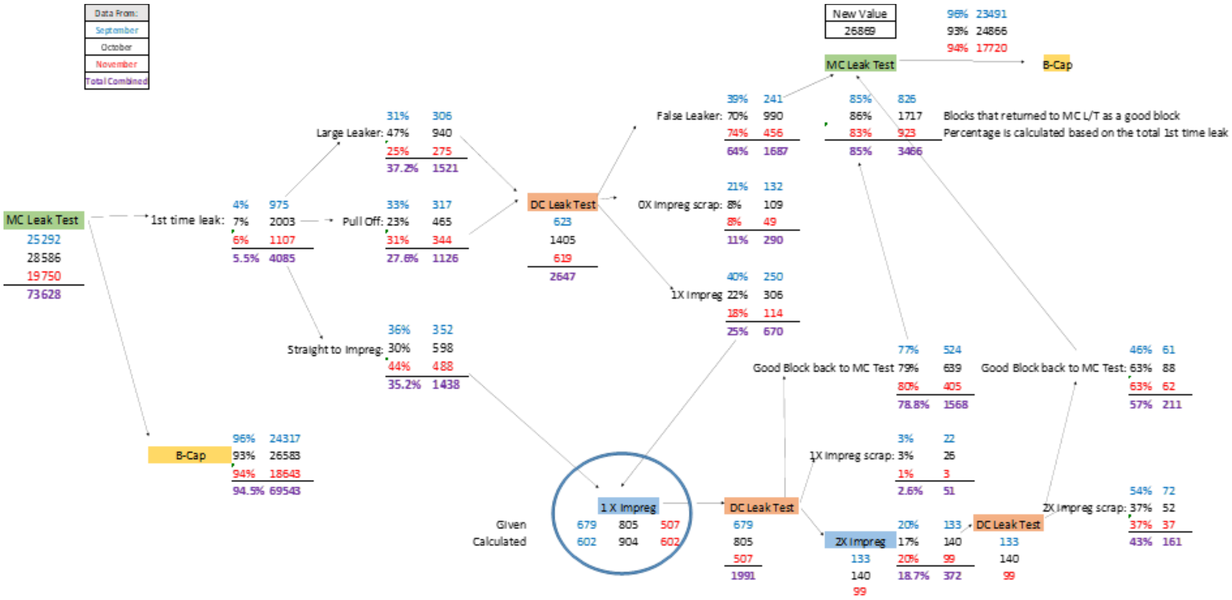


Figure 14 Analyzing Three Months of Data to Determine the Paths of an Engine Block.

4.3.2.2 Verification and Validation

In this section the verification and validation of the model is demonstrated. To ensure accuracy in simulation models, it is often advisable to use a long warm-up period and to increase the number of replications to reduce sampling error [37]. The length of the warm-up period determines the time required for the model to reach a steady-state condition, which reduces the potential for bias in the output data. A warm-up period in simulation refers to a period of time at the beginning of the simulation where the model is allowed to run without recording any output [39]. In this particular case study, a warm-up period equal to one-third (one month) of the total run time (three months) was chosen, given the complexity of the conveyor system and the significant effect of the initial conditions on the simulation output. Since the conveyor system is in operation continuously, a full month was deemed sufficient for the system to attain a steady-state condition. This duration was supported by the managers in the area. This duration allowed the model to account for any transient behavior in the system and produce accurate and reliable output data once the warm-up period was complete.

Replications are necessary since this system contains stochastic parameters with random probability distributions [58]. By increasing the number of replications, the sampling error is reduced, which leads to more accurate and reliable results [38]. The appropriate number of replications is determined based on the half-widths of the confidence intervals [39]. The half-width parameter provides information about the level of sampling error in the simulation output and is used to construct 95% confidence intervals [39]. By increasing the number of replications, the half-width can be reduced, resulting in narrower confidence intervals and more precise estimates. Controlling the number of replications and the length of the simulation run is important in experimentation, as it allows for the identification of the appropriate parameters that produce satisfactory results [56].

Table 10 displays the results obtained from the simulation model when varying the number of replications from 1 to 100. The computational time for each simulation run was recorded, along with the outputs for three measures of performance (Leak tested, 1x impregnation parts, and 1x impregnation scrap). The third column, that shows the relative computational time in percentage format, indicates the percentage relative increase in computational time compared to the time required to run the simulation with only one replication. For example, if the simulation run with one replication took 51.5 seconds, and a simulation run with 5 replications took 166.5 seconds, then the computational time percentage for 5 replications is 323%, indicating that the computational time increased by a factor of 3.24 compared to the time required for a single replication. As shown in Table 10, the half-width results for the three measures of performance decrease as the number of replications increases.

It is evident that the progress made in terms of reducing half-widths beyond 30 replications is quite insignificant, and yet the computational time keeps doubling in each line. The half widths appear to be halved between the 5th and 10th replications, as well as between the 10th and 30th replications. Up until 30 replications, the computational time seems to double with each consecutive replication increase, while the half widths are halved. However, beyond that point, although the computational time continues to double, the improvement in half widths is not as substantial.

Table 10 Number of Replications.

Replications	**Computational Time (Seconds)**	**Relative Computational Time (%)**	**Leak Tested Half-Width (%)**	**1X Impregnation Parts Half-Width (%)**	**1X Impregnation Scrap Half-Width (%)**
1 replication	51.5	100%	-	-	-
5 replications	166.5	323%	0.17%	2.00%	27.82%
10 replications	267.7	520%	0.08%	1.14%	12.51%
30 replications	672.5	1306%	0.04%	0.93%	6.78%
50 replications	1155	2243%	0.03%	0.74%	4.74%
100 replications	2437.4	4733%	0.02%	0.50%	3.01%

The results presented in Table 10 provide insights into the trade-off between the number of replications, computational time, and the accuracy of the simulation output. This information is used to identify the number of replications required for the simulation model to produce accurate and reliable results, while minimizing the computational time required to generate those results [39].

Figure 15, Figure 16, and Figure 17 were created to visually represent the relationship between the number of replications and the half-width results for the three measures of performance. Each plot displays a curve that represents the trend of the half-width results as the number of replications increases. The x-axis represents the number of replications, while the y-axes represent the computational time (left-hand side of the graph) and the half-width value for each performance measure (right-hand side). Figure 15 shows the half-width results for the number of engine blocks that were leak tested. Figure 16 shows the results for the number engine blocks that went through impregnation one time. Figure 17 shows the results for the number of engine blocks that were scraped after going through impregnation one time. By examining these plots, we can observe that as the number of replications increases, the half-width results decrease for all three performance measures.

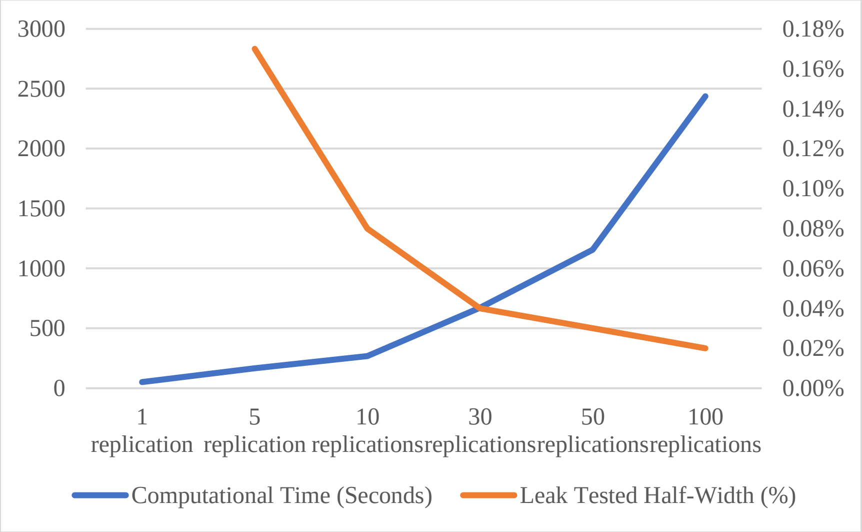


Figure 15 Leak Tested Half-Width Compared to the Computational Time.

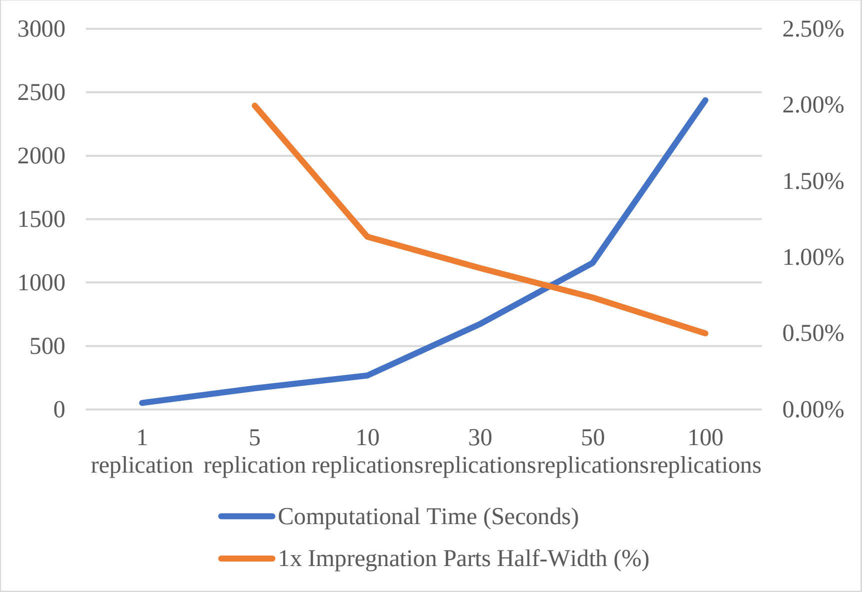


Figure 16 1x Impregnation Parts Half-Width Compared to the Computational Time.

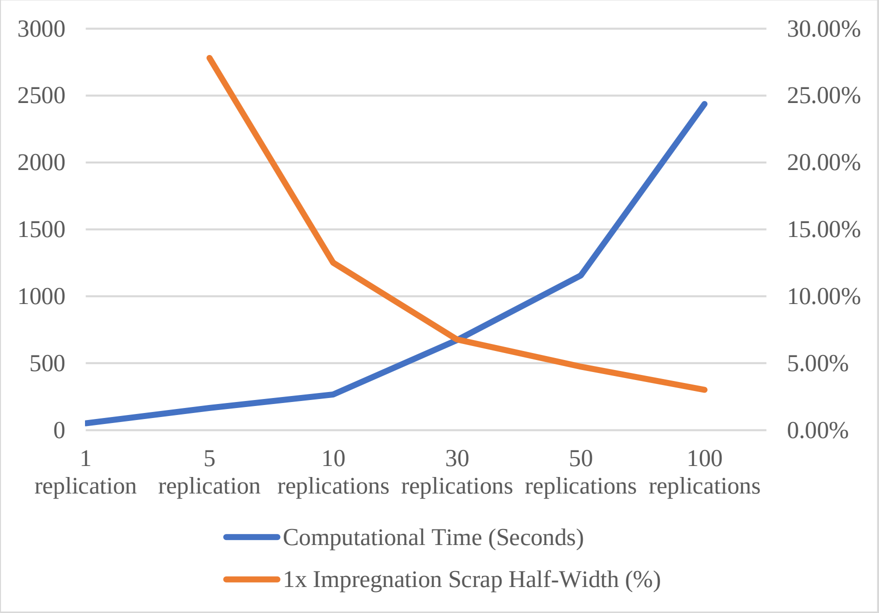


Figure 17 1x Impregnation Scrap Half-Width Compared to the Computational Time.

Based on the scales of these graphs, at the 30-replication point in the x axis the orange line slope begins to level off, while the blue line slope becomes steeper. The reverse happens before the 30-replication point, indicating a trade-off between the computational time and half-width. Therefore, 30 replications are selected for this case study as it represents the point at which replications beyond 30 would not be worth the additional computational time needed. Thus, the model was run for three months after the warmup of one month, and 30 replications were performed.

In this study, verification was performed by applying Parameter Variability-Sensitivity Analysis [111]. The Status-Label from Simio was used to track the engine blocks in the system and evaluate the output variation based on changes on the input. Status-Label displays values of specific expressions as the model runs allowing the tracking of engine numbers at any given time. Several outputs were observed, such as the number of engine blocks tested in MC Leak Test, number of engine blocks and how many times they routed through impregnation, and number of blocks considered as scrap. The results shown in the Status-Label were compared to expectations from the changes on the inputs, and the model discrepancies were adjusted until a convergence of the model to the expected results was obtained.

To validate the Baseline model, Event Validity Analysis [39] was performed, where three-months of historical data from the real system was compared to the 30 replication's results obtained from the simulation. This process was used to determine the model's accuracy and reliability in replicating the behavior of the physical system. Table 11 presents a comparison between actual data collected over a three-month period and the results obtained from a simulation run of the same duration. The table displays the percentages of engine blocks tested in the MC Leak Test that underwent various processes. For example, "1x impregnation blocks" indicates that 2.70% of the leak tested blocks received a single impregnation treatment. The relative error results, shown in Table 11, were calculated to obtain the deviation from the real outcome. To calculate relative error, the simulated value was subtracted from the real value and then the absolute value of that number was divided by the real value. When looking at the results, the Baseline model's results were found to be very close to the real data. Most of the measures of performance were accurate, within a 3-9 % range. After consulting with the managers of the leak test area, it was determined that these numbers were still indicative of typical monthly performance. To keep the confidentiality of the manufacturing plant, detailed numerical results of the verification and validation process are shown as percentages.

Table 11 Validation Results for Baseline Model.

Validation	**Real Data (%)**	**Baseline Model Data (%) ± Half- Width (%)**	**Relative Error (%)**
1x Impregnation Blocks	2.70%	2.72% ± 0.025%	4.40%
2x Impregnation Blocks	0.51%	0.56% ± 0.012%	3.93%
0x Impregnation Scrap	0.39%	0.38% ± 0.010%	6.87%
1x Impregnation Scrap	0.07%	0.07% ± 0.005%	6.64%
2x Impregnation Scrap	0.22%	0.25% ± 0.008%	3.99%
False Leak %	2.29%	2.19% ± 0.028%	8.51%
Total Leak Scrap	0.68%	0.71% ± 0.022%	1.10%

4.3.2.3 Sensitivity Analysis

In this case study, in addition to analyzing the Baseline model regarding the measures of performance determined in the Modeling phase, there is interest in testing the system under different combinations of system parameters. The system parameters, as well as their test levels were chosen in conjunction with the stakeholders to reflect current and potential realistic operating conditions. Specifically, the sensitivity of the Baseline model was tested with respect to three main parameters:

WIP Trigger: In practice, there is no clear-cut rule for when off-line WIP blocks should be put back into the line. The associate in charge of reintegrating the offline WIP blocks into the production line also has other duties to perform in the plant. As a result, the associate will reintroduce the WIP blocks whenever they have availability from their other responsibilities. In order to model this behavior more formally, a threshold of WIP that would trigger the activity was determined, called WIP Trigger. It is defined that current practices are equivalent to a WIP Trigger of 3 skids worth of engine blocks. Additionally, there was interest in whether variation of the WIP Trigger (0 and 10 skids) would influence the results.

<u>Worker Availability:</u> The worker that is responsible for the MC Leak Test cell manages the WIP in that area, loading and unloading blocks from and to that conveyor. This same worker also operates in other machines and performs other tasks during the shift. The current time that the worker is assigned to performing WIP tasks is 2 hours/shift. Managers were interested in whether this worker could perform the same tasks in less time. It was decided to make the worker available for either 1 or 2 hours.

<u>MC Leak Test Processing Time:</u> In the current state system, the MC Leak Test Processing Time is fixed to slightly below the cycle time of the line (current). There was interest in whether a delay in MC Leak Test processing time would impact the results because management believed that if its processing time is increased, it could improve the fidelity of the leak test. MC Leak Test Processing Time was used with a delay of 33%, and MC Leak Test processing time with a delay of 44%. These exact percentages were determined by the managers of this area.

A summary of the levels for each of the systems parameters that were considered for the analysis is shown in Table 12.

Table 12 Systems Parameters.

WIP Trigger (Skids)	**MC Leak Test Processing Time (Seconds)**	**Worker Availability (Hours)**
0	Current	1
3	+33%	2
10	+44%	-

The systems parameters that represent the current state of the system are: WIP Trigger of 3 skids, MC Leak Test Processing Time of slightly below the cycle time of the line (current), and Worker Availability of 2 hours/shift. A 95% Confidence Interval of the measures of performance from the 30 replications was calculated and the single-parameter experiments with each of the parameters (WIP Trigger, Worker Availability, and MC Leak Test Processing Time) was performed for the Baseline Model. The goal was to determine the effect of the variation of these parameters on the first three measures of performance (WIP B-CAP, WIP MC Leak Test, and line throughput). The other two measures of performance, Impregnation and Scrap rates, are not affected significantly by these parameters. Instead, those measures were later used to compare alternative scenarios with the Baseline model, using the current and best-case system parameter settings for each model.

The results from the single parameter experiments are shown in Table 13 and are classified based on whether they had positive, negative or no impact when comparing to the current state of the system. As the WIP around MCLT and B-CAP is lineside, it creates congestion and does not offer any strategic value or tactical utility, it is preferable to minimize its amount. In the context of the experiment, a positive impact for WIP MCLT and WIP B-CAP would mean fewer WIP blocks, while a negative impact would indicate an increase in WIP blocks. For the Line Throughput, positive impact means that the experiment resulted in more blocks produced and negative impact means fewer blocks. Positive impacts are represented in green and negative impacts are represented in red in Table 13. No impacts means that the results were within the confidence interval of the current state and thus were not significantly impacted from the change in parameters.

Table 13 Single Parameter Results for Baseline Model.

	WIP MCLT	WIP B-CAP	Line Throughput
Parameters	BL	BL	BL
WIP Trigger 0	-80%	-74%	0%
WIP Trigger 3	Current	0%	0%
WIP Trigger 10	+198%	+177%	0%
MCLT Current	Current	0%	0%
MCLT +33%	+54%	0%	0%
MCLT +44%	+1205%	0%	-1.3%
1 hour	0%	0%	0%

From the results in Table 13, the parameter WIP Trigger of 10 skids had a significant negative impact on the WIP MCLT and WIP B-CAP. Also, the parameter MC Leak Test +44% seconds had a very high negative impact WIP MCLT. As these results were significantly negative, it was decided to eliminate these two levels when performing the two-parameter experiments. The Line Throughput was mostly not impacted by varying these parameters, except a small effect when using MCLT Processing time of +44% seconds.

Once the impact of these single-parameter experiments was analyzed, a Design of Experiments was created to identify important interactions that may be missed when experimenting with one parameter at a time. The two-parameter experiments performed were: WIP Trigger x Worker Availability (Table 14), MC Leak Test Processing Time x Worker Availability (Table 15) and WIP Trigger x MC Leak Test Processing Time (Table 16).

Similarly to the single-parameter results for the two-parameter experiments, 30 replications were performed for four months of production, dropping the first month as a warm-up period. A 95% confidence interval of the results of the Baseline Model under the current operating conditions was calculated.

Table 14 WIP Trigger x Worker Availability for Baseline Model.

	WIP MCLT			WIP B-CAP			Line Throughput	
	1 hour	2 hours		1 hour	2 hours		1 hour	2 hours
	Baseline Model			Baseline Model			Baseline Model	
WIP Trigger 0	-64%	-86%		-55%	-75%		0%	0%
WIP Trigger 3	0%	Current		0%	Current		0%	Current

Table 15 MC Leak Test Processing Time x Worker Availability for Baseline Model.

	WIP MCLT			WIP B-CAP			Line Throughput	
	1 hour	2 hours		1 hour	2 hours		1 hour	2 hours
	Baseline Model			Baseline Model			Baseline Model	
MCLT Current	0%	Current		0%	Current		0%	Current
MCLT +33%	+1171%	+84%		0%	0%		-1.9%	0%

Table 16 WIP Trigger x MC Leak Test Processing Time for Baseline Model.

	WIP MCLT			WIP B-CAP			Line Throughput	
	MCLT Current	MCLT +33%		MCLT Current	MCLT +33%		MCLT Current	MCLT +33%
	Baseline Model			Baseline Model			Baseline Model	
WIP Trigger 0	-88%	0		-76%	-73%		0%	0%
WIP Trigger 3	Current	0%		Current	-72%		Current	0%

From the results in Table 14, Table 15, and Table 16, the parameter WIP Trigger 0 had a significant positive impact on the WIP MCLT and WIP B-CAP for both WIP Trigger x Worker Availability (Table 14) and WIP Trigger x MC Leak Test Processing Time experiments (Table 16). Whereas for the MC Leak Test Processing Time x Worker Availability (Table 15) experiment, the results were negatively impacted when increasing the MCLT Processing time by +33% seconds.

The Analyzing (S2) level of the 4S framework is achieved by performing input analysis, verifying and validating the model, and performing sensitivity analysis. Once the model was capable of analyzing, the next level of capability, Predicting (S3), could be added.

4.3.3 Applying Predicting (S3)

In the third level of the 4S framework, Predicting, the simulation model is able to predict the outcome of how the physical system will operate and produce with a given set of inputs. To add this level to our simulation model, what-if scenarios were determined by varying the model inputs and structure. In this case study two alternative scenarios are presented and the results from the single-parameter and two-parameter experiments that provide predictions are discussed. Comparisons within scenarios, as well as across scenarios, relative to the current state of the system are presented. For these experiments, the measures of performance focused on are WIP B-CAP, WIP MCLT, and Line Throughput.

After several brainstorming sessions with the stakeholders, including managers, engineers and associates, two improvement layouts that were realistic and applicable were determined, labeled as Layout 1 (L1) and Layout 2 (L2).

Layout 1: The first what-if scenario considered in this study is titled Layout 1 (L1). The only difference, when comparing to the Baseline Model, is that MC Leak Test is no longer the only Master Leak Test; instead, DC Leak Test is also considered as a Master Leak Test. In other words, this scenario trusts DC Leak Test results and does not perform retesting in MC Leak Test. When a block passes DC Leak Test, it goes straight to the next process, B-CAP, instead of going through MC Leak Test again. The initial decision to have MC Leak test as the only master leak tester was managerial, not technical; experts in this area believe that DC Leak Test results are at least as accurate as MC Leak test results and hence this change would have no impact on test quality. This layout prevents (what management believes to be) unnecessary retests in MC Leak Test. A visual representation of Layout 1 is shown in Figure 18.

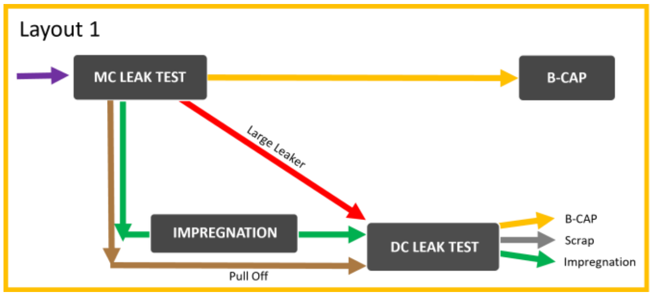


Figure 18 Layout 1 Model.

Layout 2: The second improvement scenario considered in this study is titled Layout 2 (L2). This layout considers that all blocks that fail MC Leak Test should be tested in DC Leak Test, before deciding their destination, which is either they need impregnation, or they should go back to MC Leak Test, or if they are to be scrapped. This scenario allows DC Leak Test to catch false leakers from MC Leak Test, preventing unnecessary impregnation. It also limits spending unnecessary time in lineside WIP, but it still uses MC Leak Test as the Master Leak tester of the system, so the number of false negatives that move on to B-CAP is no worse than in the Base Line model. A visual representation of Layout 2 can be found in Figure 19.

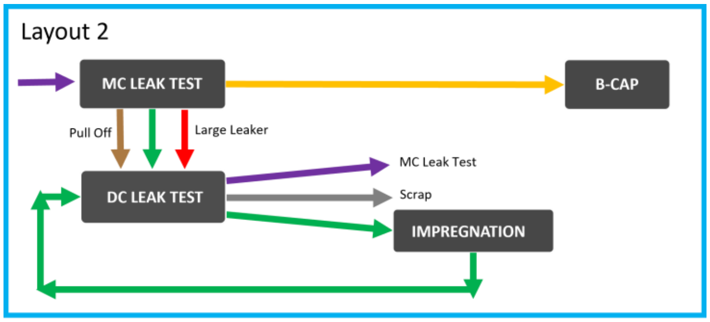


Figure 19 Layout 2 Model.

Once the alternative scenarios were defined, simulation results for Layouts 1 and 2 were obtained and compared with the Baseline Model results. The results were classified based on whether they had positive, negative or no impact, as explained in the previous section. Table 17 shows the results for each of the measures of performance that were impacted by the variation in levels of the system parameters under each scenario. These results are presented relative to the Baseline model results, where falling within the original confidence interval was considered "no impact" or 0%. It can be observed that the parameter WIP Trigger of 10 skids had a significant negative impact on the WIP MCLT and WIP B-CAP, when comparing Layout 1 and 2 to the current values from the Baseline model. As explained in the previous section, the WIP trigger of 10 skids and the MC Leak Test +44% seconds were eliminated to perform the two-parameter experiments since these results were significantly negative and hence no further experimentation was necessary to advise against using these levels.

Table 17 Single Parameter Results.

	WIP MCLT			WIP B-CAP			Line Throughput		
Parameters	BL	L1	L2	BL	L1	L2	BL	L1	L2
WIP Trigger 0	-80%	-92%	-87%	-74%	-55%	-72%	0%	0%	0%
WIP Trigger 3	Current	0%	0%	0%	0%	0%	0%	0%	0%
WIP Trigger 10	+198%	+188%	+194%	+177%	+197%	+172%	0%	0%	0%
MCLT Current	Current	0%	0%	0%	0%	0%	0%	0%	0%
MCLT +33%	+54%	0%	+43%	0%	0%	0%	0%	0%	0%
MCLT +44%	+1205%	0%	+1003%	0%	0%	0%	-1.3%	0%	-1.2%
1 hour	0%	0%	0%	0%	+174%	0%	0%	0%	0%
2 hours	Current	0%	0%	0%	0%	0%	0%	0%	0%

Once the single parameter experiments were performed for both alternative layouts, the two-parameter experiments were executed, and the results were evaluated.

Table 18 shows the results from the experiment varying WIP Trigger and Worker Availability. In Layout 1, the Worker Availability of 1 hour in combination with the WIP Trigger 0 and 3 generated a negative impact in the WIP B-CAP, which implies that only 1 hour is not enough to manage WIP around the B-CAP machine area. In both Layout 1 and 2, the worker availability of 2 hours in combination with WIP Trigger 0 skids had a positive impact in the WIP

MCLT and WIP B-CAP. However, WIP Trigger of 0 skids is not realistic because the worker is not actually going to manage WIP for just a couple of engine blocks, when they have other more pressing responsibilities. Table 19 shows the results from the experiment varying MC Leak Test Processing Time and Worker Availability. The results show negative impact when MC Leak Test processing time is +33% seconds and the worker is only available 1 hour/shift, indicating that this combination is detrimental to the process. Table 20 shows the results from the experiment combining WIP Trigger and MC Leak Test Processing Time. As expected, the WIP Trigger 0 results in fewer blocks in WIP around MC Leak Test and B-CAP.

Table 18 WIP Trigger X Worker Availability.

	WIP MCLT			WIP B-CAP			Line Throughput	
	1 hour	2 hours		1 hour	2 hours		1 hour	2 hours
	Baseline Model			Baseline Model			Baseline Model	
WIP Trigger 0	-64%	-86%		-55%	-75%		0%	0%
WIP Trigger 3	0%	Current		0%	Current		0%	Current
	Layout 1			Layout 1			Layout 1	
WIP Trigger 0	-89%	-92%		+97%	-57%		0%	0%
WIP Trigger 3	0%	0%		+157%	0%		0%	0%
	Layout 2			Layout 2			Layout 2	
WIP Trigger 0	-70%	-87%		-50%	-68%		0%	0%
WIP Trigger 3	0%	0%		0%	0%		0%	0%

Table 19 MC Leak Test Processing Time X Worker Availability.

	WIP MCLT			WIP B-CAP			Line Throughput	
	1 hour	2 hours		1 hour	2 hours		1 hour	2 hours
	Baseline Model			Baseline Model			Baseline Model	
MCLT Current	0%	Current		0%	Current		0%	Current
MCLT +33%	+1171%	+84%		0%	0%		-1.9%	0%
	Layout 1			Layout 1			Layout 1	
MCLT Current	0%	0%		+200%	0%		0%	0%
MCLT +33%	0%	0%		+211%	0%		0%	0%
	Layout 2			Layout 2			Layout 2	
MCLT Current	0%	0%		0%	0%		0%	0%
MCLT +33%	+1157%	+41%		0%	0%		-1.7%	0%

Table 20 WIP Trigger X MC Leak Test Processing Time.

	WIP MCLT		WIP B-CAP		Line Throughput	
	MCLT Current	MCLT +33%	MCLT Current	MCLT +33%	MCLT Current	MCLT +33%
	Baseline Model		Baseline Model		Baseline Model	
WIP Trigger 0	-88%	0	-76%	-73%	0%	0%
WIP Trigger 3	Current	0%	Current	-72%	Current	0%
	Layout 1		Layout 1		Layout 1	
WIP Trigger 0	-94%	-87%	-58%	-51%	0%	0%
WIP Trigger 3	0%	0%	0%	0%	0%	0%
	Layout 2		Layout 2		Layout 2	
WIP Trigger 0	-89%	0%	-72%	-72%	0%	0%
WIP Trigger 3	0%	+38%	0%	-63%	0%	0%

Under the current process, blocks can go through impregnation zero times (0x impregnation), one time (1x impregnation) and two times (2x impregnation). Blocks are not supposed to go through impregnation for a third time (3x impregnation). However, since the blocks are not tracked, there is no way to know if that actually happens or not. The simulation results predicted that it could happen and probably does, which implies that the current routing rules are having unexpected and undesirable consequences. The results from the Baseline model (see Table 21) indicate that there is an average of 0.2% of blocks that go through impregnation for a third time. While it is possible to impregnate a block for the third time in the Baseline model, Layouts 1 and 2 do not allow it to happen, as the results show zero 3x impregnation blocks.

Table 21 Impregnation Results.

Simulation Data	**Baseline Model**	**Layout 1**	**Layout 2**
1x Impregnation Blocks	82.7%	-3%	-50%
2x Impregnation Blocks	17.1%	-12%	-53%
3x Impregnation Blocks	0.2%	-100%	-100%
Total Impregnation Blocks	100%	-4%	-51%

There was also interest in looking at the scrap related to this process. A scrap block is classified by how many times it went through impregnation, (0x,1x, or 2x). When looking at the scrap results (see Table 22) of Layout 1, it did not vary much from the Baseline model; however, Layout 2 had a significant change. When analyzing the scrap results for Layout 2, there is a significant increase in 0x impregnation scrap and fewer 1x impregnation and 2x impregnation blocks being scrapped when compared to the Baseline model, which means that the blocks were getting scrapped earlier in the process. This would reduce the cost of unnecessary impregnation.

It is important to emphasize that the reduction in impregnation cost obtained from Layout 2 does not affect the quality of impregnation. The impregnation cost that is being eliminated refers to being more selective about what goes through impregnation and scraping blocks earlier in the process. That is, scrapping blocks that are highly unlikely to be repaired through further impregnation. Although Layout 2 shows better improvements on the 1x and 2x impregnation scrap blocks, the total leak scrap resulted to be higher by 5% when compared to the Baseline Model. However, the additional cost generated by this increase in the number of blocks being scrapped is only a fraction (less than 5%) of the total impregnation savings that Layout 2 provides.

Table 22 Impregnation Scrap Results.

Simulation Data	**Baseline Model**	**Layout 1**	**Layout 2**
0x Impregnation Scrap Blocks	55%	-2%	+155%
1x Impregnation Scrap Blocks	10%	-4%	-50%
2x Impregnation Scrap Blocks	35%	-14%	-82%
Total Leak Impregnation Scrap	100%	-7%	+5%

The three simulation models (Base Line, Layout 1, and Layout 2) allowed the capture of the time a block spends in the Leak Test area. These times were input back into the VSM to have a measure of the total delay caused by the Leak Test area. The current state model (BL) provides that the Leak Test area accounts for 6.2% of the total lead time of the system. The total lead time of the current state of the line was then compared with the results from Layout 1 and Layout 2. In Layout 1 the total lead time in the machining area increases by 2%. On the other hand, with Layout 2 the total lead time decreases by 5%. The value of using the combination of VSM and DES simulation is that it helped identify the problem area and calculate lead times of the Leak Test area, and the machining line as a whole.

The Predicting (S3) level of the 4S framework is achieved by creating alternative scenarios, examining the results and confirming that they can accurately predict what would happen in the real system when the alternative scenarios are applied. Once the model was capable of predicting, the next level of capability, Prescribing (S4), could be added.

4.3.4 Applying Prescribing (S4)

In the last level of the 4S framework, Prescribing, a simulation optimization approach is employed to identify the most effective course of action. In this case study, the simulation optimization approach used in Simio was the add-In OptQuest, which is utilized to generate new scenarios with varying control values to search for an optimal response value [113]. This approach enables the identification of the best solution by iteratively adjusting the input variables until the desired outcome is achieved [114].

OptQuest automates the process of setting property values, starting replications, and retrieving results in Simio. During the optimization run, OptQuest sets the simulation by resetting the run and changing the values of the input properties to those identified by the Add-In. Then, it starts to run replications of the first scenario. Once the scenario is completed, OptQuest retrieves the response values needed for the objective function and uses its algorithms to determine the next set of inputs to create [39]. It then uses these inputs to run the next scenario and repeats the process until it reaches a specified stopping condition, or the user stops the session [115]. When OptQuest is not used, the user has to manually input the combinations of control values that need to be tested, which can be a tedious and time-consuming process [113]. However, with OptQuest, the user only needs to specify the minimum and maximum values for each input control along with the increment size [39]. OptQuest then uses these values to systematically generate scenario combinations by varying the input control values between their defined minimum and maximum ranges in steps determined by the increment size [39]. This makes the simulation optimization process more efficient and less prone to errors caused by manual input of control values [115].

The control values for this experiment are the same ones mentioned previously in Table 12: WIP trigger, MC processing time, and worker availability. The selection of these controls involves tradeoffs that need to be considered carefully [114]. The WIP trigger control is important for ensuring that the system does not become overloaded with WIP. Setting it too low can slow down the process and reduce the efficiency of the system. However, setting it too high can result in a backlog of work that could ultimately slow down the process as well. Therefore, a balance must be struck between keeping the WIP trigger low enough to maintain efficiency while avoiding overload. Worker availability is another important control that affects the efficiency of the system. Having more availability generally leads to increased productivity. However, the tradeoff is that more availability costs more money. Therefore, the selection of this control needs to consider the tradeoff between the increased productivity provided by the worker having more availability versus the cost of it. The MC leak test processing time control is related to the accuracy of the leak test. Slower leak test processing times can result in a lower percentage of false leakers. However, increasing the processing time can lead to a reduction in overall efficiency. Therefore, the selection of this control needs to consider the tradeoff between the fidelity of the leak test and the impact on processing times. Table 23 outlines the minimum value, maximum value, and increment size of each control variable utilized. The WIP trigger will range from 0 to 180 blocks, incrementing by 18 blocks, which corresponds to the capacity of a skid. The MCLT processing time will range from 45.7 to 65.7 seconds, increasing by 5 seconds, based on the management's choice. Worker availability will range from 1 hour to 2 hours, increasing by 1 hour. Although schedules were used to establish worker availability, Simio does not enable scheduling as a control for OptQuest. Therefore, separate OptQuest runs with the two schedules will be conducted to examine the impact of varying worker availability scenarios. The OptQuest will be executed separately for the Baseline, Layout 1, and Layout 2 models, as it is provided in Table 24. After each run, the optimal outcome will be determined and compared across all three models to identify the best result.

Table 23 OptQuest Controls.

Minimum Value	**Controls**	**Maximum Value**	**Increment Size**
0	WIP Trigger (blocks)	180	18
45.7	MCLT Processing Time (seconds)	65.7	5
1	Worker Availability (hours)	2	1

Table 24 OptQuest Plans.

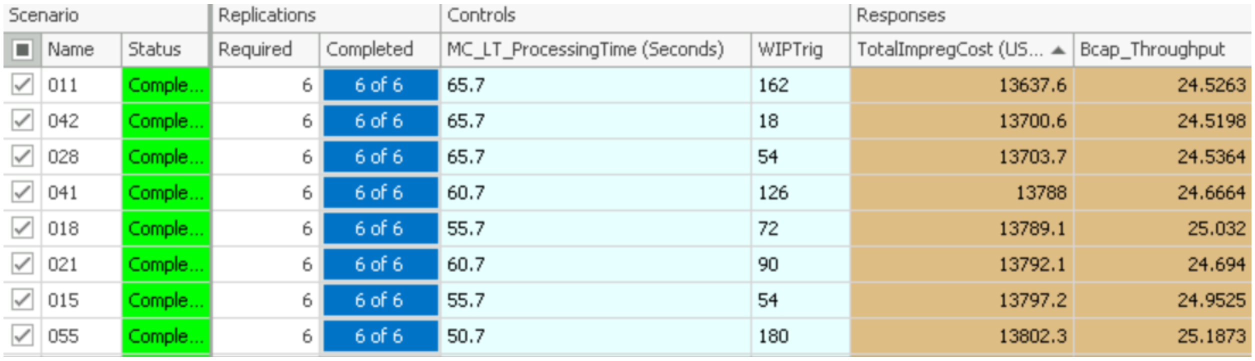
Model	**Worker Availability 1h**	**Worker Availability 2h**
Baseline	BL_optquest_1h	BL_optquest_2h
Layout 1	L1_optquest_1h	L2_optquest_2h
Layout 2	L2_optquest_1h	L2_optquest_2h

The total cost of impregnation for an engine block is $6.1, which includes impregnation, conditioner, and catalyst fluids. The goal of the optimization problem is to reduce the overall cost associated with impregnation. The objective function is calculated by multiplying the maximum impregnation value that undergo through impregnation one time, two times, and three times, and then multiplying that sum by $6.1 to obtain the total cost of impregnation for the three-month simulation period. The total impregnation cost, the throughput of the system, and the total amount of false leaker blocks are expressed in the model as a response, as demonstrated in Table 25. The goal is to minimize the total impregnation cost, while maximizing the system's throughput, and limiting the amount of false leakers.

Table 25 Responses of the Model.

Responses	**Expression**
Total Impregnation Cost	(1ximpreg.maximum + 2ximpreg.maximum + 3ximpreg.maximum)*6.1
BCAP Throughput	(B_Cap.Processing.NumberEntered / Time.Now)

After the completion of the OptQuest runs, the total impregnation cost column was sorted in ascending order and the smallest value was selected for each model. An example of the OptQuest results in Simio is shown in Figure 20.



Scenario			Replications		Controls		Responses	
■	Name	Status	Required	Completed	MC_LT_ProcessingTime (Seconds)	WIPTrig	TotalImpregCost (US... ▲	Bcap_Throughput
☑	011	Comple...	6	6 of 6	65.7	162	13637.6	24.5263
☑	042	Comple...	6	6 of 6	65.7	18	13700.6	24.5198
☑	028	Comple...	6	6 of 6	65.7	54	13703.7	24.5364
☑	041	Comple...	6	6 of 6	60.7	126	13788	24.6664
☑	018	Comple...	6	6 of 6	55.7	72	13789.1	25.032
☑	021	Comple...	6	6 of 6	60.7	90	13792.1	24.694
☑	015	Comple...	6	6 of 6	55.7	54	13797.2	24.9525
☑	055	Comple...	6	6 of 6	50.7	180	13802.3	25.1873

Figure 20 OptQuest results from BL_optquest_1h model.

Table 26 displays an overview of the OptQuest results, showcasing the optimal set of controls for all the models. It is evident from the table that both Layout 1 and Layout 2 yielded lower impregnation costs compared to the Baseline model. The optimal set of controls, leading to the minimal impregnation cost, is found in Layout 2. This configuration includes a worker availability of 1 hour, a MCLT processing time of 65.7 seconds, and a WIP trigger of 72 engine blocks. The corresponding cells in Table 26 are highlighted in blue to indicate this configuration. The changes from Layout 2 were recommended to our industry partners to be implemented.

Table 26 OptQuest Results.

	Controls		**Responses**	
	MCLT Processing Time	WIP Trigger	Total Impregnation Cost	BCAP Throughput
BL_optquest_1h	65.7	162	$13,637.60	24.53
BL_optquest_2h	45.7	18	$13,778.90	25.18
L1_optquest_1h	45.7	162	$13,342.70	25.16
L1_optquest_2h	55.7	108	$13,296	25.21
L2_optquest_1h	65.7	72	$6,669.33	24.60
L2_optquest_2h	65.7	54	$6,714.07	24.98

By systematically exploring various configurations and sorting the results, OptQuest assists in finding the optimal solutions quickly and efficiently. In the specific case discussed, OptQuest facilitated the selection of the most cost-effective layout and control parameters for impregnation, leading to significant savings compared to the baseline model. The ability to analyze and compare different scenarios using a simulation optimization approach to achieve the Prescribing level (S4) of the 4S framework enhances the overall performance and efficiency of the system being modeled. In this last level of capability, the simulation model offers valuable insights into the optimal operation or configuration of a system to support decision-making. Since the model now exhibits all four levels of capability from the 4S simulation framework, it can be considered a fully capable model.

The last level, Prescribing, is achieved when a simulation model is used to evaluate the system and provide insight into its optimal operation or configuration to support decisions. The Prescribing capability is an advanced feature of simulation modeling that can be particularly useful for decision-making purposes. It can help decision-makers identify the most effective course of action to achieve their goals and provide insights into the potential outcomes of various intervention strategies.

4.4 Chapter Summary

While there are existing frameworks for building simulations, there is currently no comparable framework for precisely defining the different levels of capability. In this research, the 4S simulation framework was proposed: Modeling (S1), Analyzing (S2), Predicting (S3), and Prescribing (S4), which represent the increasing utility and capability of a simulation model. To develop the 4S framework of simulation, two separate searches were conducted to gather literature, including books and scholarly articles. The definitions of simulation were collected and analyzed, and six main characteristics were identified and labeled as C1 through C6. The next step involved referencing back to the selected books and articles to determine which characteristics were explicitly mentioned in the definitions. Similar characteristics were then combined, resulting in four categories for the 4S framework: S1, S2, S3, and S4. To label these categories appropriately, word clouds were created to identify the most relevant and frequently used terms. Action verbs were preferred as category labels to represent capabilities within the framework.

A simulation case study in an automotive manufacturing engine plant that fulfills all the 4S levels of simulation capability was showcased. In the first level, Modeling (S1), a DES model that represented the real system (Baseline Model) was created. In the second level, Analyzing (S2), the metrics were determined, a design of experiments was created of the current system, and the results were analyzed. In the third level, Predicting (S3), two potential sets of changes in the process (Layout 1 and Layout 2) were proposed and the outcome of the proposed changes was predicted. In the fourth level, Prescribing (S4), a simulation optimization technique was employed which allows for the search of an optimal response value, continuously modifying input variables until the desired outcome is attained.

When looking at the results, the Baseline Model predicted that it is possible to have 3x impregnation in the process, which is something that is not supposed to happen. It also showed that the WIP around MC Leak Test is higher than WIP in B-CAP. Finally, it showed that under the current configuration it is possible to moderately increase the time of MC Leak test, but changing the WIP trigger, Worker Availability, or a major increase to MC Leak Test time could have very negative consequences. Results from Layout 1 show that 3x impregnation does not happen. Also, under this layout, 1 hour worker availability per shift is not enough to input WIP from B-CAP back into line. Layout 2 also eliminates 3x impregnation, but more importantly it saves the most annual impregnation cost, by scraping defective blocks earlier in the process and avoiding unnecessary impregnation. This layout increases the total scrap, but this cost is far lower than the savings from impregnation. It was recommended to the industry partners to implement Layout 2.

Chapter 5

Contribution 2: When is a Simulation a Digital Twin? A Systematic Literature Review[2]

5.1 Introduction

In this chapter, the current application of DT is investigated, by using classification to delineate the differences and similarities between simulation and DT. Additionally, this chapter includes a discussion of a future agenda for both researchers and practitioners. Clarifying the differences between the capabilities of a DT and simulation is necessary to achieve effective and accurate DT implementations. For instance, [14] identified distinctions between a model and a DT; however, numerous simulation-based models continue to be referred to as DTs. According to [13], simulation can be a component of a DT, but not every simulation is a DT. Simulation and DT are distinct technologies with unique benefits that can provide important insights to a system, therefore, they should be classified accordingly, and it is crucial to understand and highlight their differences to prevent misunderstandings. Thus, using an SLR approach, this research intends to:

- Identify whether the current modelers and researchers are consistently applying DT or if they are calling a simulation a DT.
- Highlight the main differences in capability between DTs and simulations.
- Classify DT applications according to capabilities and provide a summary of technology and industrial application fields.

---

[2] This chapter accepted for publication in special issue of Manufacturing Letters with the proceedings of the North American Manufacturing Research Conference (NAMRC) 51.

- Identify the implementation challenges of simulation-based DTs and recommend future research directions.

Many papers regarding DT have been published since Grieves presented the concept [116], including nine systematic literature reviews (SLR). None of these nine works achieve the objectives of this research. [16] conducted a comprehensive review focusing on analysing DT in terms of its concepts, technologies, and industrial applications, and presented recommendations related to the different lifecycle phases of the DT. [117] analyzed twenty-two papers with an emphasis on examining the status of DT applications in the construction industry. Similarly, [118] also performed a SLR in the construction industry but with the goal of identifying the countries or regions that are active drivers of DT adoption and assessing its impacts. They found that the majority of the developed countries, such as the UK, US, Australia, and Italy have the highest number of researchers contributing to the driving forces for the adoption of DT in the construction industry. [119] conducted a SLR to investigate the temporal evolution of research fronts and emerging research trends in the field of physical internet and DT in supply chain management. [29] identified categories of barriers to DT implementation and provided a conceptual model that prescribes how these categories in the process industry affect each other. [93] reviewed DTs from an engineering and business innovation perspective and identified future perspectives for DT. [120] reviewed papers related to DT in product design and development, classifying them into conceptual design, detailed design, design verification, and redesign. [121] reviewed DT technologies and implementation challenges in several domains and applications in engineering and beyond. [122] provided a detailed explanation of how the DT reported in the manufacturing literature are structured and how they function. A comparison of these nine papers can be found in Table 27.

Overall, these existing SLRs focus on providing an overview of established definitions of DTs, classifying the DTs in terms of application, components and technologies, suggesting further research, and highlighting the current gap in the literature in specific application areas. Although such reviews are extensive and detailed, no SLR or other study in the literature has looked at the relationship between DT and simulation. Note that this is a natural relationship, as these two techniques have some similar capabilities and objectives. In fact, several papers reviewed herein, presented in section 5.3.5, build simulations and call them DT but it is unclear if these simulations have the full capabilities usually associated with DT.

One of the issues facing the widespread adoption of DT is the lack of an agreed-upon taxonomy. In the world of simulation, terminology has become standard and accepted. The definitions are clear, and the vocabulary used is well-known. Whereas, in the world of DT the engineering community is still in the process of forming this shared language which is something that has happened organically for some topics such as simulation, but not as of yet for DT.

The use of simulation combined with a DT is common which creates an immense misconception about classifying a simulation model as a DT and vice versa [98]. There are several applications where a traditional simulation serves a valuable purpose, but to call a simulation a DT is not accurate. The 4R framework [96,97] and the 4S framework provided in Section 4.2 is used to classify the works reviewed herein as either simulation and/or DT, based on the capabilities that they possess. The classification approach presented in this research will contribute toward creating a consensus about when a simulation model is a DT and when it is not.

Table 27 Comparison of Systematic Literature Reviews of DT.

Ref.	Time scope of initial papers	Num. of considered papers	Keywords	SLR purpose
[117]	2010-2020	22	{“digital twin” or “digital twins” or “virtual counterpart” or “digital replica” or “virtual twin”} AND {“construction” or “construction industry” or “construction engineering” “construction management” or “construction engineering and management”}	To examine the application of DT in respective lifecycle phases in the construction industry.
[119]	2013-2021	518	{“digital twin” OR “physical internet” OR “hyperconnected”} AND {“supply chain” OR “logistics” OR “manufacturing” OR “procurement” OR “inventory” OR “transport” OR “purchasing” OR “storage assignment” OR “order picking”}	To provide the emerging research trends and future research directions in the Physical Internet /DT-Supply Chain Management field.
[29]	2016-2020	47	{“digital twin” OR “digital shadow” OR “device twin” OR “device shadow” OR “digital alias” OR “virtual twin” OR “virtual shadow”} AND {“teel” OR “glass” OR “pharmaceutical” OR “ceramic” OR “stone” OR “clay” OR “metal” OR “chemical” OR “food” OR “beverage” OR “textile” OR “wood” OR “paper” OR “process” OR “manufacturing”} AND {“barriers” OR “obstacles” OR “enable” OR “driver”}	To review the enablers of and barriers to the implementation of DTs in the process industry.
[118]	2018-2023	58	{“digital twin” OR “virtual counterpart” OR “digital replica” OR “virtual twin”} AND {“construction” OR “construction industry”}	To determine the region of authors actively probing into the drivers of DT adoption and assess the impacts on their construction industry.
[93]	2015-2019	123	{“digital twin” OR “virtual twin” OR “cyber twin”}	To examine the state-of the-art research articles from their engineering and business innovation perspectives.
[120]	2011-2020	60	{“digital twin” OR “digital twin product design” OR “digital twin product development” OR “digital master” OR “digital shadow” OR “digital avatar”}	To identify the current states of DT research focusing on product design and development through summarizing industrial cases.
[121]	2017-2021	18	{“digital twin”}	To review the DT technology and its implementation challenges in relevant domains and applications context.
[122]	2020	54	{“digital twin” OR “digital twinning”} AND {“definition” OR “process model” OR “system architecture” OR “conceptual model” OR “conceptual architecture”}	To analyze how the DT systems are structured, and assess its potential applicability to Architecture, Engineering, Construction, and Operations use cases.
[16]	2003-2019	176	{“digital twin” OR “digital twins” OR “digital replica” OR “product avatar” OR “virtual twin”}	To analyze DT from the perspective of concepts, technologies, and industrial applications, and recommend future research.

5.2 Methodology: Systematic Literature Review

This study is grounded in an SLR approach that focuses on minimizing bias by applying systematic methods that are documented in advance with a protocol [123]. To examine the literature within the defined scope this study adapted the methods used by [16,29]. Moreover, a three-stage process was used to select the academic journal papers, eliminate publications that were not closely related, and analyze the content of the related papers. This three-step process is composed of: (a) literature search, (b) literature selection, (c) review process.

The SLR methodology employed in this study is illustrated in Figure 21. Considering the main objective of investigating the current state of the art concerning DT and simulation levels of capability, two research questions (RQ) were posed:

- RQ1: Are simulation models that do not have any DT characteristics being referred to as DTs?
- RQ2: Do existing implementations of DT have all of the capabilities to fit into a 4R category?

RQ1 aims to investigate whether in the existing literature authors are creating DTs or just applying simulation instead and calling it a DT. RQ1 becomes of fundamental importance because simulation and DT are different technologies, therefore, they present different properties and capabilities. RQ2 aims to investigate if the existing DT applications present characteristics and capabilities of DTs as defined by the 4R framework.

In the first step of the SLR methodology, five academic search engines were used to gather relevant literature: Web of Science, Engineering Village, Science Direct, IEEE, and ASME. These five database sources were used to ensure that an acceptable number of research papers were captured and used in this study, since they include publications in a variety of fields. An initial and comprehensive search was conducted using the keywords with appropriate Boolean operators to select publications with the term “digital twin” in the title and “simulation” in the abstract. This analysis only includes papers in English and published in peer-reviewed scientific journals. The search was restricted to journal articles because they undergo a more thorough peer-review process, ensuring the accuracy of the information published. Conference papers, technical reports, and theses/dissertations may not have the same level of scrutiny as academic journal articles, and their peer-review process can vary greatly, which is why they have not been included in this analysis. While there are some conferences, such as the Winter Simulation Conference, which could have been included due to their size and importance in the field, the SLR methodology does not allow for arbitrary inclusion or exclusion of specific conferences. Therefore, no conference papers were included in this SLR.

After combining the papers obtained from each database, the total number retrieved was 795 publications: Web of Science (350 papers), Engineering Village (333 papers), Science Direct (73 papers), IEEE (38 papers), and ASME (1 paper). To ensure that this SLR is comprehensive, rigorous, and representative of the available evidence, the decision was made to focus on databases that return a higher number of papers. By utilizing these databases, the likelihood of identifying all relevant studies and a satisfactory number of studies to perform significant statistical analyses increases. Additionally, databases with a higher number of papers returned are more likely to include a diverse range of studies from various disciplines, regions, and publication sources. Therefore, an exclusion criterion was applied to eliminate databases that returned less than 100 papers, resulting in the use of only two databases: Web of Science (350 papers) and Engineering

Village (333 papers), which together contain 683 publications. After removing duplicates, the total number of papers was 469.

Then, a detailed, critical, and comprehensive examination was conducted to identify and select the most appropriate publications for this research. This was executed in the second step of the SLR, which consisted in two parts: first reading the title and abstract of each paper and selecting the papers that were related to a practical application of DT and then reading the full paper to further eliminate any other unrelated papers. For the inclusion or exclusion criteria, only publications that focused on DT application and presented a case study in their full text were used in this research. Any papers outside the application range with no mention of DT implementation in their title or abstract, such as conceptual papers, literature reviews, frameworks, methodologies, surveys, theories, and mathematical models were excluded from this analysis. After analyzing the title and abstract, a total of 236 papers were discarded, resulting in 233 potentially relevant publications for the full text analysis. During the full text review, 113 unrelated papers were excluded, resulting in 120 relevant and unique publications to be thoroughly analyzed and categorized.

In the last step of the SLR, the set of publications was selected and the articles were categorized according to the levels of capability of DT and according to the levels of capability of simulation observed. The DT level was determined by reviewing the case studies reported by the authors and applying the definitions of each capability level (or R) presented in the 4R framework. The level of simulation was applied to categorize the papers according to the levels of simulation (or S) defined in earlier. Overall, the journals with the most publications in the final set of 120 were: Journal of Manufacturing Systems (12), International Journal of Advanced Manufacturing

Technology (9), International Journal of Computer Integrated Manufacturing (8), Sensors (6), and IEEE Access (5).

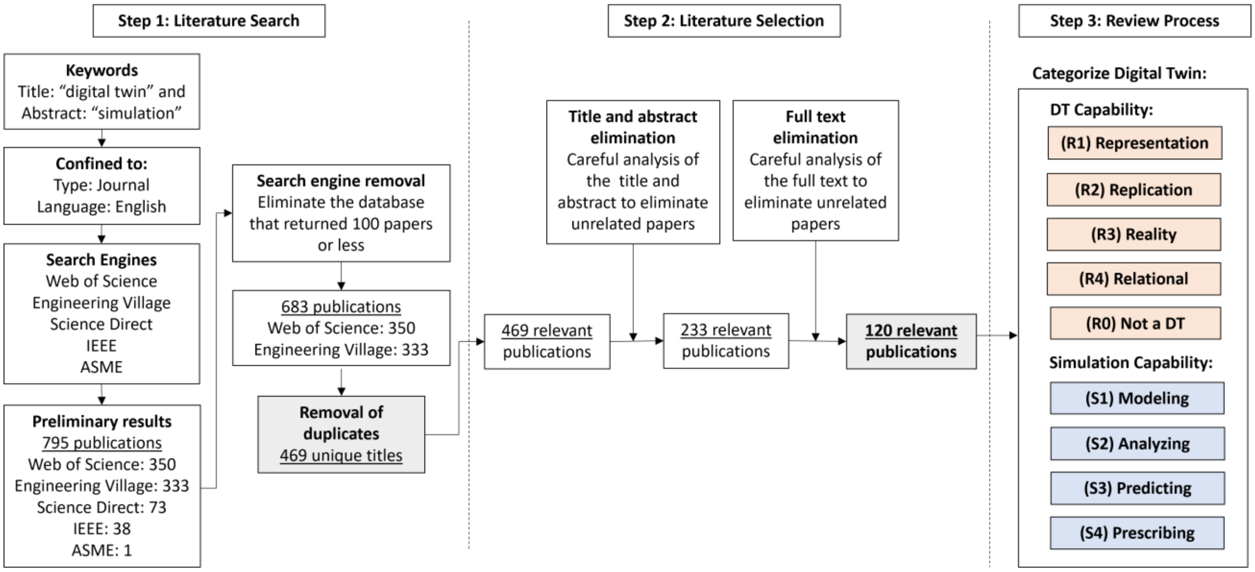


Figure 21 SLR Methodology.

A breakdown of the papers by year of publication is shown in Figure 22, along with the three steps of the SLR with the purpose of demonstrating how many papers were disregarded in each of these three steps. As can be observed, the development of DT was rather slow between 2009 and 2018. However, there was a significant increase in publications from 2019 to 2022, indicating a development in the concept of DTs. Note that the relevant academic publications employed in this work (from step 3 of the SLR) only cover publications from 2018 to 2022, but they exhibit the same trends as the total number of publications first acquired in steps 1 and 2. Also, the year 2022 refers to documents published (possibly online) and indexed in the databases by March 29th, 2022, when the search was conducted.

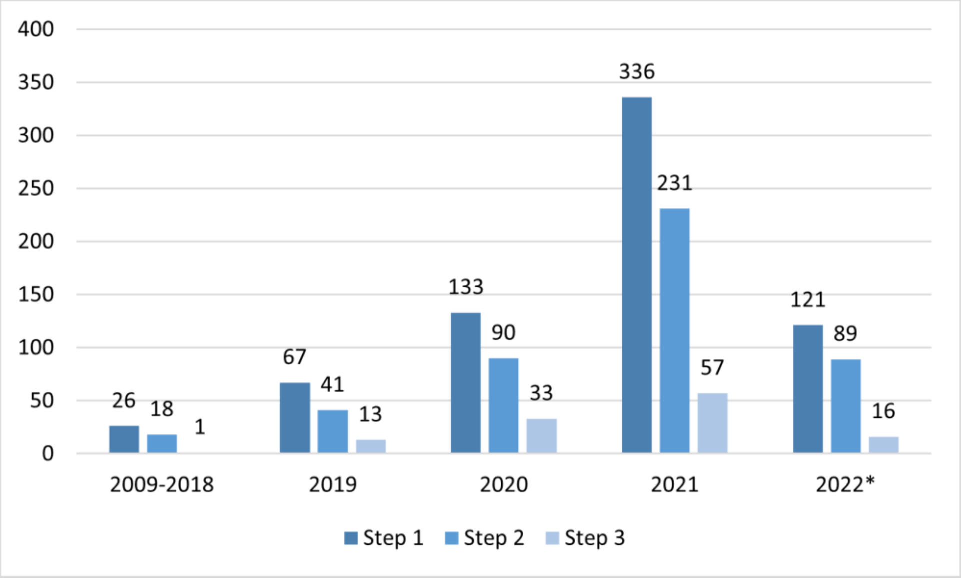


Figure 22 Reviewed Publications by Year of Publication.

5.3 Capability Classification

A classification of DT and simulation levels of capabilities was conducted for each paper selected through the SLR methodology described above. This classification provides information for understanding whether the current literature is truly applying DT or if it is using simulation in place of DT. Each publication was categorized within the levels of 4Rs and 4Ss. The 4R's are the levels of capability and maturity of a DT are presented as: (R1) Representation, (R2) Replication, (R3) Reality, and (R4) Relational. Whenever a publication did not achieve any levels of the 4R's it would be classified as R0, which means that the model described did not show the characteristics what would qualify it as a DT. The 4S's assesses the levels of capability and utility of a simulation model. These four levels are presented as: (S1) Modeling, (S2) Analyzing, (S3) Predicting, and (S4) Prescribing. The results of this categorization of articles provide insight to the state of current literature in appropriately labeling DTs. The 4R and 4S classification results are shown in Figure 23 and Figure 24, respectively.

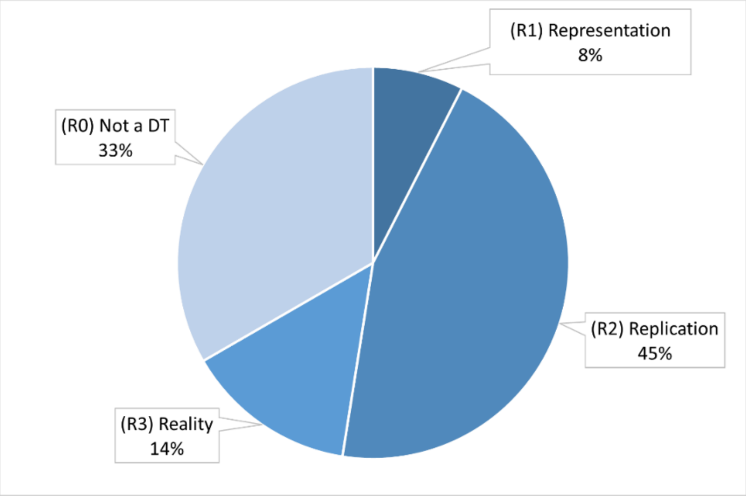


Figure 23 Summary of 4R Classification.

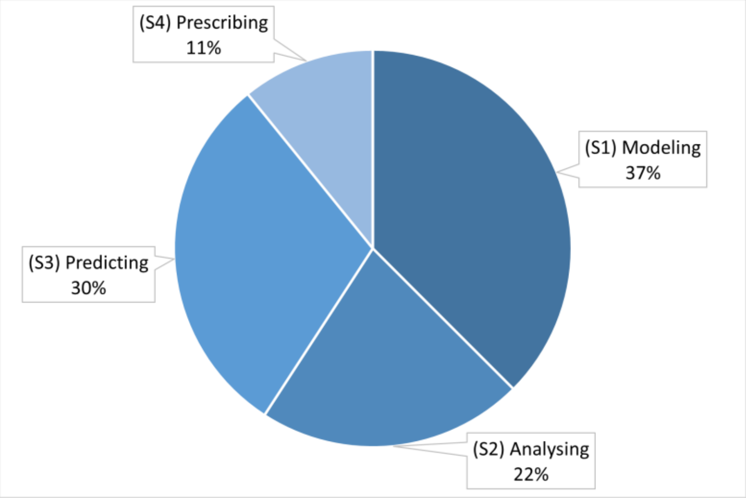
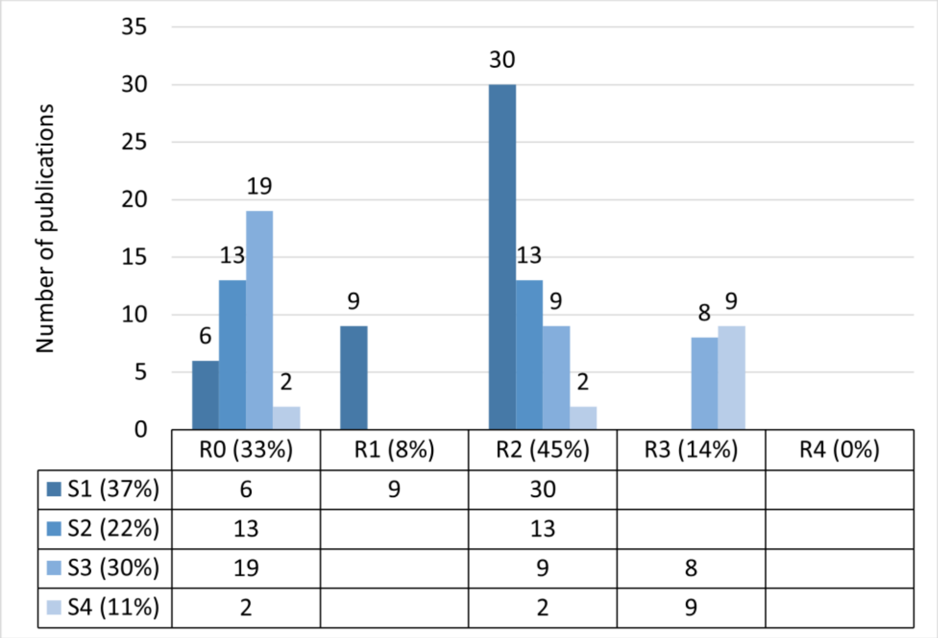


Figure 24 Summary of 4S Classification.

As observed in Figure 23, one third of the papers that claimed to present a DT did not have any of the 4R capabilities, so they should not be called DT. Of the papers classified as presenting a DT, 8% of the papers only reached R1, while the most common level of DT capability achieved by the applications of DT in the literature was R2 (45%). The papers that reached R3 (14%) tend to be very detailed models. Most of the articles reviewed lie in the early stages of DT levels of capability. There were no publications found that reached the R4 level. This indicates the infancy of DT implementations and the large gap still left to fill. These results validate the hypothesis that some researchers are incorrectly calling their models a DT when it is actually a simulation model. With respect to the simulation capability classification, as seen in Figure 24, the majority of the papers (37%) achieved the capability level of modeling (S1), 22% achieved the analyzing level (S2), followed by 30% achieving the predicting level (S3). Only a few publications (11%) create a fully capable model that can analyze results, predict outcomes, and prescribe optimal solutions, achieving the prescribing level (S4).

Looking at the combination of the 4Rs and 4Ss (Figure 25), the combination R2-S1 is the most common, followed by R0-S3 and the relative number of publications achieving the R2 level of capability decreases with an increase in the level of simulation capability. Analyses on these classifications are conducted in the following subsections. The results are presented from highest to lowest levels of capability.



	R0 (33%)	R1 (8%)	R2 (45%)	R3 (14%)	R4 (0%)
■ S1 (37%)	6	9	30		
■ S2 (22%)	13		13		
■ S3 (30%)	19		9	8	
■ S4 (11%)	2		2	9	

Figure 25 Combined Analysis of 4Rs and 4Ss Levels of Capabilities.

5.3.1 Relational Classification (R4)

A DT is categorized into the Relational (R4) level of capability, in accordance with the 4Rs framework, when the physical and virtual systems are in sync and data is bidirectional in the available literature that feature DT applications that achieve the highest level of capability. None of the articles reviewed achieved this level of complexity and capability.

5.3.2 Reality Classification (R3)

A DT is categorized into the Reality (R3) level when it is used to evaluate hypothetical scenarios with the goal of incorporating the results of the virtual runs into the actual system in order to optimize it. Only 14% of the articles utilized in this study were able to employ the R3 level of capability. Although they each presented a DT that offers real-time data collection and produces optimization results, they are unable to develop solutions on their own and be self-learning. Regarding the levels of simulation capability, eight of these models achieved level S3, and nine achieved level S4. An overview of these papers and the year they were published is shown in Table 28 and Table 29.

5.3.2.1 Reality (R3) and Prescribing (S4)

The papers in this category achieved the Prescribing level of capability, which means they can assess data, predict outcomes, and make recommendations. The nine papers classified as R3S4 [124–132] are those that are closest to the last level of capability (R4) of the 4R framework, using all levels of simulation capability. They developed DT models that collect real-time data and generate optimal outcomes. Some propose using different technologies to achieve this. For instance, virtual reality (VR) is used by [131] to connect historical and real-time data and analyze various scenarios and IoT is used to control and link processes between the physical and digital worlds. Table 28 presents the DT applications classified into the R3S4 levels of capability.

Table 28 Digital Twin Applications Classified as R3S4.

Refs.	Proposed Idea for the R3S4 papers	Year
[128]	Design and development of a DT for a case study of a pharmaceutical company.	2020
[132]	Describe a DT approach that supports rapid reconfiguration of the line for changeovers.	2020
[124]	Propose an improved multi-fidelity simulation-based optimization method with ordinal transformation and optimal sampling.	2020
[125]	Construct a DT of workers in different scenarios to realize the deep fusion of physical trajectory and spatial virtual electric field distribution.	2021
[130]	Introduce the DT technologies to optimize the gas exchange system in terms of performance and manufacturing efficiently and precisely.	2021
[131]	Propose a closed-loop dynamic air cargo loading DT, integrating a cargo load plan optimization simulation, multi-dimensional immersive VR system, IoT, and real-time sensors.	2021
[129]	Propose a novel DT joint optimization approach for warehousing in large-scale automated high-rise warehouse product-service system.	2021
[127]	Propose a direct possibility of testing and debugging advanced logistics algorithms using a DT outside the production line.	2021
[126]	Propose a DT clamping force control approach to improve machining accuracy of thin-walled parts.	2022

5.3.2.2 Reality (R3) and Predicting (S3)

The eight articles classified as R3S3 present a DT that can assess data and predict outcomes under alternative scenarios, but they are unable to prescribe solutions. For instance, [133–137] presented a DT model that reflects and monitors its physical counterpart, clearly using real-time data collection, and dynamically providing optimal results. Although some authors [138–140] claim that their DT shows bidirectional data interaction, their case studies are not sufficiently detailed to support this assertion. Their DTs can predict behaviors, but they are not always linked to their physical system, preventing them from becoming autonomous. Table 29 lists DT applications classified into the R3S3 levels of capability.

Table 29 Digital Twin Applications Classified as R3S3.

Refs.	**Proposed Idea for the R3S3 papers**	**Year**
[141]	Describe the methodology and application of a DT in a manufacturing plant in China.	2020
[133]	Propose a novel DT-based intelligent cooperation framework to facilitate the implementation of algorithms in unmanned aerial vehicle swarm.	2020
[142]	Provide an overall framework manufacturing and operation and maintenance integration of a complex product based on DT.	2021
[138]	Present a method of cutting parameter optimization on the basis of the construction of the DT of a Computer numerical control (CNC) machine tool.	2021

[136]	Provide a multi-dimensional DT dedicated to product lifecycle of the constant velocity joint.	2021
[137]	Realize the dynamic scheduling of DT job-shop based on edge computing.	2021
[139]	Propose a framework of DT-based industrial cloud robotics for industrial robotic control.	2021
[140]	Describe the modeling and implementation of an integrated system that consists of a real material handling system and its DT.	2021

5.3.3 Replication Classification (R2)

The bulk of the publications (45%) used DT as a virtual replica at the R2 level. These studies presented a DT that can connect with real-time data and replicate the same outputs as the physical system, but they lack the ability to independently research alternate scenarios and find solutions to issues. Thirty of these papers attained level S1, thirteen did so at level S2, nine reached level S3, and two attained level S4. An overview of these studies and the year they were published is shown in Table 30, Table 31, Table 32, and Table 33.

5.3.3.1 Replication (R2) and Prescribing (S4)

Only two publications, classified as R2S4, made full use of simulation capabilities. Using their proposed DT, [143] replicate the system, link real-time data, and assign tasks to the actual scenario. The physical system is virtualized, inferences are drawn about it, future events are predicted, and the best course of action is suggested. Similarly, the simulation model-to-real process interface developed by [144] enabled the integration of both, transforming the virtual model into a representation of the real system and replicating the same outputs as the real system. Table 30 lists the DT applications classified as R2S4.

Table 30 Digital Twin Applications Classified as R2S4.

Refs.	**Proposed Idea for R2S4 papers**	**Year**
[143]	Discuss an object-oriented event-driven simulation as a DT of a flexible assembly cell coordinated with a robot alongside humans.	2019
[144]	Analyze the use of the DES as a DT in a non-automated process.	2020

5.3.3.2 Replication (R2) and Predicting (S3)

Both the Predicting (S3) level of simulation capability and the Replication (R3) level of DT capability are attained by nine works. Although they developed models that can accurately simulate the behavior of the actual system and produce results that are identical to those of the real system, make prediction, and consider different scenarios, they did not incorporate any optimization techniques to enhance system performance. A system that connects real-time data from the physical to the virtual world is the main topic of most articles. Some writers, like [145,146], connected historical data from the real process to be automatically entered into the virtual system, employing it as real-time data. On the other hand, real-time synchronization between the physical and the digital model was established by [147–150]. Several other authors provide the technologies used to establish these connections. For instance, [151,152] use a DT model that is continuously updated with real-time data gathered from sensors. IoT was used by [153] to gather and record real-time data from physical space and their model can imitate behaviors and anticipate the system with accuracy. Table 31 provides a list of the DT applications classified into R2S3 levels of capability.

Table 31 Digital Twin Applications Classified as R2S3.

Refs.	Proposed Idea for R2S3 papers	Year
[145]	Develop a health monitoring and prognosis of permanent magnet synchronous motor by creating an intelligent DT in MATLAB/ Simulink.	2019
[146]	Propose a method that uses VR to simulate complex tasks with the DT modeling of workers' cognitive reactions to various scenarios.	2020
[154]	Propose a framework of DT-driven production management system to support a cyber-physical system of production workshop.	2020
[147]	Propose a DT that guides the selection of antenna strips at a base station.	2020
[148]	Demonstrate feasibility of DT implementation under real conditions of a production plant of aluminum components.	2020
[149]	Present a prototype of a DT manufacturing system design platform.	2021
[151]	Introduce a five-dimension DT for a machine in the job-shop and propose a DT-enhanced dynamic scheduling methodology.	2021
[152]	Develop a DT framework for real-time logistics simulation, which can predict potential logistics risks and accurate module arrival time.	2021
[153]	Propose a real-time data-driven energy behavior model of equipment for creating the DT model of energy-efficient manufacturing system.	2022

5.3.3.3 Replication (R2) and Analyzing (S2)

Both the Replication (R2) level of DT capability and the Analyzing (S2) level of simulation capability are attained by thirteen studies. The models used by these authors can examine the system under different parameter settings and provide analysis of the results, but they are unable to predict outcomes or make recommendations.

Some authors [155–157] only demonstrate how they linked real-time data to their DT to replicate the system. [158] focuses on how to precisely describe the system in the virtual environment by using the data fusion approach. [159] proposes the use of ML and AI in their DT methodology indicating they would reach a higher level of the DT capability, but their case study only provided connection of real-time data to replicate the system into a virtual space. Other authors, such as [160–166], connect PLC and sensors to develop the virtual model but it is dependent on the original system to function. Table 32 presents the DT applications classified into R2S2 levels of capability.

Table 32 Digital Twin Applications Classified as R2S2.

Refs.	Proposed Idea for R2S2 papers	Year
[161]	Propose the implementation of DT approach as part of a wider cyber-physical system to enable the optimization of the planning and commissioning of human-based process.	2018
[162]	Propose a co-simulation and communication architecture between DT and VR software.	2019
[159]	Develop a DT architecture reference model to enable the context-aware product family design optimization process in a cost-effective manner.	2020
[155]	Present a DT implementation with an inverse method using strain gauges as load sensors.	2020
[157]	Introduce a generic architecture for DT establishment in smart manufacturing.	2020
[156]	Develop a framework for implementing a DT and provide a case study as proof of concept.	2021
[160]	Develop a DT that mimics appearance, structure, behavior, state, kinematics, and dynamics of the physical system to support online commissioning.	2021
[163]	Propose a DT empowered mobile edge computing architecture.	2021
[167]	Propose a DT-enabled VSM approach for Small and Medium Enterprises based on an Efficiency Validate Analysis simulation.	2021
[158]	Present an assembly precision analysis method based on a general part DT model.	2021
[164]	Develop a DT-driven Human-robot collaboration system that measures the motions of a worker and simulates the working progress and physical load based on digital human technology.	2021
[166]	Describe the development of a control system using the DT methodology for a gas system.	2022

[165]	Illustrate the conception, realization, and validation of an original solution for the DT of a material extrusion 3D printer.	2022

5.3.3.4 Replication (R2) and Modeling (S1)

The majority of the studies that fell under the R2 level (30) just executed the first simulation capability, S1. Instead of conducting experiments or making optimized predictions, these authors just concentrate on creating a virtual model that accurately represents what occurs in a physical system. Most of these authors [168–176], develop a DT model architecture that uses real-time data, replicating the real system into a virtual world with no experimentation or analysis. Some authors [26,177–187] focus on employing sensors to establish a real-time connection between the two. Others provide alternative approaches to this connectivity. For instance, [188] integrates the simulation program and robotic arms layouts using augmented reality, whereas [189] uses VR to synchronize the digital replica with the physical system. Other authors, such as [190–194], focus on validating and verifying their DT models. Besides that, some authors [195,196] mention that they have bidirectional connectivity between the virtual and physical, but their case study shows otherwise. Table 33 provides a summary of these papers.

Table 33 Digital Twin Applications Classified as R2S1.

Refs.	Proposed Idea for R2S1 papers	Year
[189]	Develop a novel DT prototype to analyze the requirements of communication in mobile networks supported remote surgery.	2019
[174]	Present a potential solution, based on the Industrial IoT middleware, that implements a fully dual-way synchronization between the real and virtual worlds.	2019
[179]	Introduce a DT tool to support the lightweight design of assemblies in composite material.	2020
[178]	Present a multi-layer architecture that provides the infrastructure required for a DT within the cyber-physical production systems paradigm.	2020
[177]	Describe the development of a real-time DT of a wound rotor induction machine using a precomputed finite element model fed with online measurements.	2020
[188]	Present a methodology of using augmented reality technique to create a DT of robotic arms.	2020
[191]	Propose a semantic conceptual framework for industrial process modeling in the context of DT.	2021
[194]	Present a machining DT capable of real-time adaptive control of intelligent machining operations.	2021
[195]	Propose a methodology design using model-driven engineering that strives toward being flexible and generic.	2021

[175]	Use DT of low voltage side of distribution transformers to calculate in real time the waveforms of their medium voltage sides.	2021
[180]	Propose the first DT framework that can implement both real-to-sim and sim-to-real information flow to achieve better robustness for deploying robots in challenging environments.	2021
[181]	Demonstrate a novel proof of concept Industry 4.0 production system which lays the foundations for future research in DT technologies, process optimization and manufacturing data analytics.	2021
[168]	Develop a DT of corridor traffic that leverages real-time data streams to model the current traffic state and provide dynamic feedback on traffic and environmental performance measures.	2021
[176]	Propose a multimedia knowledge-based bridge health monitoring using DT.	2021
[26]	Propose an approach to develop a DT of production systems in order to optimize the planning and commissioning process.	2021
[182]	Present a pipedream end-to-end simulation engine for real-time modeling and state estimation in natural/urban drainage networks.	2021
[183]	Describe the implementation of a DT emulator of an automated mechatronic modular production system that allows for exchanging near real-time information with the physical system.	2021
[169]	Propose a model-based systems engineering construction method of shop floor DT.	2021
[192]	Demonstrate a novel multisource model-driven DT system for producing a precise and real-time simulation of a robotic assembly system.	2021
[184]	Propose a new DT design concept based on external service for the transportation of the Automatic Guided Vehicles.	2021
[193]	Apply DT in the deformation of the sheet metals based on the novel incremental bending process.	2021
[170]	Propose a DT model that assists in the online /remote programming of a robotic cell by creating a 3D digital environment of the real-world.	2021
[196]	Develop a DT architecture able to optimize productivity in the context of Controlled Environment Agriculture applications.	2021
[173]	Focus on a discrete manufacturing workshop layout optimization based on DT.	2021
[185]	Develop and test a DT prototyping platform for architectural design development that allows users to engage simultaneously with the physical and digital worlds.	2022
[186]	Propose an automatic traffic modelling method to model the real-world traffic and to recreate the modeled traffic in the simulated 3D environment.	2022
[171]	Develop a physics-based DT model to provide valuable information for both transportation and bridge	2022
[187]	Propose the use of existing data standards and web technologies to modeling and development of DT ships.	2022
[172]	Build a three-dimensional representation of a 12-person meeting room and the surrounding area using modeling, physics simulation, and rendering capabilities of Unity 3D.	2022
[190]	Develop a DT system for thermal characteristics.	2022

5.3.4 Representation Classification (R1)

Nine of the publications (8%) employed in this study achieved just the R1 level, adopting the DT as a virtual representation. These DT models share the trait of digitally representing and comprehending the behavior of physical systems. Even though these papers presented a case study of a DT that offers real-time data collection, they fall short in terms of analyzing different scenarios, producing optimization results, or achieving autonomy. Regarding the levels of simulation capability, all of these articles achieved level S1. An overview of these papers and the year they were published is shown in Table 34.

5.3.4.1 Representation (R1) and Modeling (S1)

Both the Representation (R1) level of DT capability and the Modeling (S1) level of simulation capability are attained by nine studies. The models used by these authors can represent the physical system in a virtual world, but they are unable to examine multiple scenarios, provide analysis of the results, predict outcomes, or make recommendations. Some authors [197–201], concentrate on offering the real-time data connection but don't offer any analysis or experiment outcomes. Others [202–205] focus on developing a cloud platform for data collection and storage. Table 34 lists the DT applications classified as R1S1.

Table 34 Digital Twin Applications Classified as R1S1.

Refs.	Proposed Idea for R1S1 papers	Year
[197]	Propose an approach for the online diagnostic analysis of power electronic converters utilizing real-time, probabilistic digital twinning.	2020
[204]	Create a DT for an experimental assembly system based on a belt conveyor system.	2020
[202]	Present a mechanism for constructing a unified manufacturing process, and results of an integrated multiscale simulation of an injection molding process.	2021
[198]	Discuss further development of the DT concept as well as potential inferences based on the data collected.	2021
[205]	Propose a modeling approach for building a DT for friction stir welding based on a sensor-based numerical simulation.	2021
[201]	Study the real-time data acquisition method that fuses the acquisition node and the assembly process of a spacecraft and establish the virtual-real mapping process around the data space.	2021
[200]	Describe the concept of a DT creation, which could be used as part of a centralized ground traffic control system in the airport.	2021
[199]	Construct a new intelligent logistics distribution management system based on machine vision and visual sensor image processing technology to respond to the shortcomings of the system.	2021
[203]	Propose a novel DT-enabled collaborative data management framework for metal systems.	2022

5.3.5 Not a Digital Twin (R0)

After the Replication category, the R0 group has the second-highest number of articles presented. The authors in the R0 group (33%) simply create a simulation model and call that a DT. Their models do not offer any real-time data collection or bidirectional data connections. Instead, they use historical data as input, and they vary within the levels of simulation capability. Six of these papers attained level S1, thirteen did level S2, nineteen got to level S3, and two reached level S4. An overview of these papers and the year they were published is shown in Table 35, Table 36, Table 37, and Table 38.

5.3.5.1 Not a DT (R0) and Prescribing (S4)

Only two publications, classified as R0S4, demonstrated that their models fully used all the simulation capabilities. [206,207] do not synchronize real-time data to their models, which makes it purely a simulation. They present a complete simulation model that can analyse alternative scenarios, predict outcomes, and prescribe optimization solutions. Table 35 summarizes these two papers.

Table 35 Digital Twin Applications Classified as R0S4.

Refs.	**Proposed Idea for R0S4 papers**	**Year**
[206]	Present a DT framework for assembly systems with compliant parts fusing sensors with deep learning and simulations.	2020
[207]	Demonstrate how performance losses induced by variable times can be recovered using DT.	2020

5.3.5.2 Not a DT (R0) and Predicting (S3)

The S3 level of simulation capability is attained by nineteen articles. Although the authors in this category describe their model as a DT, the virtual model created has more characteristics of a very detailed simulation model of the object with no real-time connection. Some authors [208–224] attempt to obtain real-time data from the system but then use it as historical data. According to [225], the DT must produce predictions using real-time data, what-if analyses, and/or ML, but it is unclear whether they accomplished that. Even though [226] provided a DT composed by a virtual model using DES, an AI tool to provide accurate input data to the model, and a DT interface using dashboards, they do not appear to use real-time data at all. Table 36 presents a summary of these papers.

Table 36 Digital Twin Applications Classified as R0S3.

Refs.	Proposed Idea for R0S3	Year
[223]	Develop a DT of a mango fruit to simulate its thermal behavior throughout the cold chain, based on the environmental temperature conditions.	2019
[216]	Introduce the DT of a real-world Electric Vehicle by modeling the mobility based on a time series behaviors of electric vehicles to evaluate the charging algorithm.	2019
[222]	Construct a three-layer super-network model to provide quantitative research for data among heterogeneous subjects in digital twinning.	2019
[208]	Propose individualized locator adjustments as a new method to improve the geometrical quality of assemblies.	2019
[209]	Present a methodology to calculate the Remaining Useful Life of machinery equipment by utilizing physics-based simulation models and DT concept, in order to enable predictive maintenance for manufacturing resources using Prognostics and health management techniques.	2019
[217]	Present a framework for implementing the DT-driven approach for developing ML models.	2020
[210]	Present the design methodology, mathematical analysis, simulation study, and validation of a DT for fault diagnosis in photovoltaic installations.	2020
[211]	Show that high-fidelity non-destructive inspection data and new developments in numerical modeling can be combined to create a DT for mitigation of void formation in composite parts.	2020
[212]	Propose a DT framework for the health management of reusable spacecraft.	2020
[213]	Implement a non-deterministic DT framework for the health management of fatigue critical structures	2020
[221]	Report distributed fiber optic sensors embedded in Inconel alloy components to validate numerical models of additive manufacturing process.	2020
[225]	Use DT to predict and analyze material removal process.	2020
[214]	Develop higher performance Atkinson cycle gasoline engine and explore its fuel-saving potential on series hybrid electric vehicle.	2021
[218]	Present a pressure-driven sieve tray column model using design correlations for the calculation of pressure drop and both the liquid and vapor holdup.	2021
[226]	Propose a DT to aid in the operational planning of a medium-sized Fast Fashion company.	2021
[219]	Develop a DT to allow a real-time mapping of the patient flow in order to create a sustainable and dynamic vaccination center.	2021
[215]	Propose the use of artificial neural networks in DES models to determine the current distribution of each event outcome.	2021
[224]	Present two approaches for digital twinning in the context of the forecast of power production by photovoltaic panels.	2022
[220]	Apply the DT concept to develop an advanced simulation technology based on the technical definition of operation and procedures.	2022

5.3.5.3 Not a DT (R0) and Analyzing (S2)

Thirteen papers reached the S2 level of simulation capability. Even though these authors [227–234] simulate, experiment with different parameters, and evaluate performance, they make no mention of gathering real-time data and connecting it to the virtual environment. Some studies [235,236] created a platform for simulation-optimization and conducted several experiments, employing sensors to gather data, but there is no direct connection to the virtual space. Other authors [237,238], use DES and call it a DT. They collect real-time data but do not automatically connect it to the simulation software. [239] use Monte Carlo simulation and DES to create scenarios and implement the results later in the real system. Table 37 presents a summary of these articles.

Table 37 Digital Twin Applications Classified as R0S2.

Refs.	Proposed Idea for R0S2 papers	Year
[227]	Present a data-driven DT system for automatic process applications by integrating virtual modeling, monitoring, diagnosis, and optimized control into a cooperative architecture.	2019
[228]	Use DT for multiscale simulation of a single carbon fiber filament with controlled boundary conditions to study the effect of the input graphitic crystal microstructure and dislocations between microstructures.	2020
[229]	Present a methodology to produce a multi-fidelity DT model for a wind turbine blade.	2020
[233]	Present a patient-specific finite element model approach, focusing on tibial plateau fractures, to enhance biomechanical knowledge to optimize surgical trauma procedures and improve decision-making in postoperative management.	2021
[235]	Develop a simulation-optimization platform on the performance and emissions of engine to optimize the fuel consumption and emissions.	2021
[230]	Develop a DT containing multiple dimensions for the autoclave to analyze the characteristics of autoclave under different conditions.	2021
[234]	Build a 5-dimensional DT and present a DT-based optimization strategy that considers machining efficiency and aerodynamic performance.	2021
[236]	Present an intelligent fault diagnosis framework for machinery based on DT and deep transfer learning.	2021
[239]	Employ a DT to optimize time-based maintenance policy in the mining industry.	2021
[237]	Address blending control strategies prior to bitumen extraction and provide a pathway to incorporate geological variation into decision-making processes throughout the value chain.	2021
[238]	Propose a co-simulation approach for engineering DTs that are used to train Bayesian Networks for fault diagnostics at equipment and factory levels.	2022
[231]	Develop a simulation model and propose a method for evaluating the introduction of automation in a pharmaceutical laboratory.	2022
[232]	Develop an integral modeling approach for manufacturing processes in order to assess their status and performance.	2022

5.3.5.4 Not a DT (R0) and Modeling (S1)

Six papers' simulation capability is limited to the S1 level. Some works [240–243], develop a digital representation of the physical system but do not provide any type of real-time synchronization between the virtual and physical spaces. For instance, [244] describe a simulation model of a griding process using Markov chains. [245] mention that their model has a direct data connection from the physical to the virtual, but they are not clear presenting how that information flows. Table 38 presents a summary of these papers.

Table 38 Digital Twin Applications Classified as R0S1.

Refs.	Proposed Idea for R0S1 papers	Year
[244]	Address the construction of DT using hidden Markov models for the industry 4.0.	2019
[245]	Present a new approach to develop DT of a helicopter dynamic system.	2019
[243]	Present the results of parametric numerical modeling of a DT of a stir welding process tool.	2020
[240]	Present a DT of a vertical transportation system, focusing on modeling and using it to evaluate the system condition and corrective solutions.	2020
[241]	Present a hybrid modelling method to monitor performance of control stage systems.	2020
[242]	Present the formulation of a DT by the enhancement of an optical model.	2021

5.4 DT Terminology and Supportive Technologies

Although the notion of DT is defined in several scholarly publications, the bigger goal of having a universal definition of DT has not yet been met. Instead, the current trend simply expands in scope by adding more definitions, leading to confusion in the field that encourages the usage of alternative technologies under the DT term. Although each new concept is logically justified, none of them has been shown to be more suitable than others and most researchers have failed to make a distinction between DT and simulation models. Numerous concepts are proposed without certain crucial components, which contributes to the definition's ambiguity and prolongs the quest for a comprehensive and agreed-upon definition. To find some association between the definition of DT used by each author and the level of capability of their application of DT, the source of the definition of DT used in each of the academic papers within this work was compiled. Some authors offer a list of numerous definitions, while others only offer one, and yet others offer none at all. To clarify, the DT definition gathered was the definition that was first mentioned in each work.

The top four sources of DT definitions utilized in the papers within the corpus of 120 pertinent publications are displayed in Figure 26. As can be observed, DT was not defined in 32 papers (26%). The majority of the publications used definitions from Grieves et al. [5,83,246] (17%), Tao et al. [6,77,87,247](10%), and Glaessgen et al.[248] (6%). The other 41% of the publications displayed inconsistent use of a singular definition (1-3 times per manuscript).

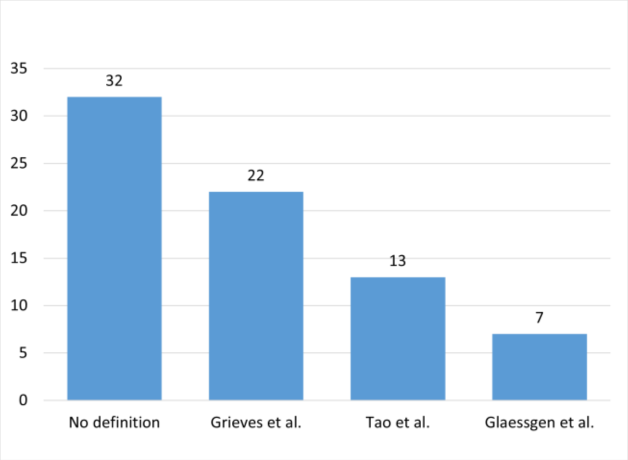


Figure 26 Top Four Sources of DT Definitions.

Numerous terms were found for how a DT is referred to in the literature. Sixty-nine percent papers referred to it as "Digital Twin," whereas 31% of the remaining scholarly articles referred to DT under a different designation summarized in Table 39.

Table 39 Summary of Other Denominations to DT.

Other Digital Twin Denominations
Digital Twin Assisted
Digital Twin Based
Digital Twin Driven
Digital Twin Enabled
Digital Twin Emulator
Digital Twin Job Shop Based
Digital twin Simulation
Digital Twinning
Data Driven Digital Twin
Interactive Digital Twin
Intelligent Digital Twin
Real Time Digital Twin
Simulation Digital Twin
Simulation Based Digital Twin
Virtual Reality Based Digital Twin

Furthermore, the industry sectors in which the DT applications considered in the SLR were applied is considered. Figure 27 illustrates the categorization of these papers according to the industry in which they are used. The manufacturing industry has the most studies, followed by energy and automotive.

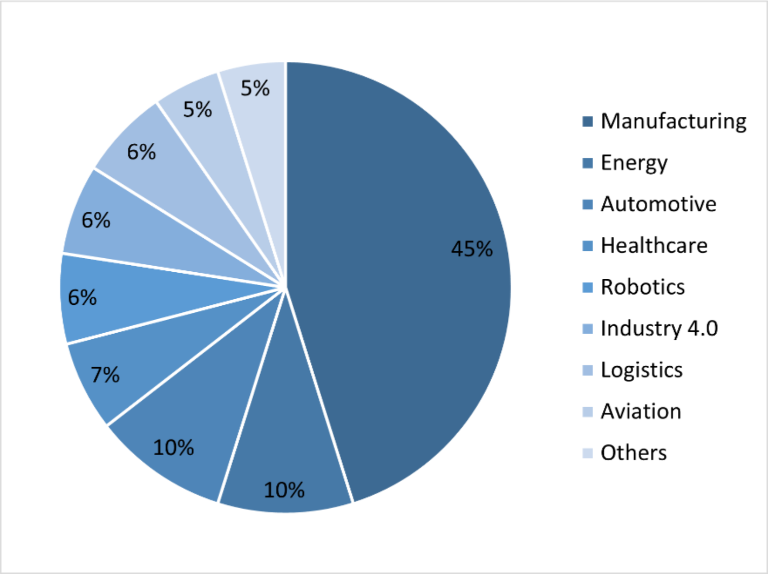


Figure 27 Industry Sectors.

For DT applications, numerous technologies—such as big data, IoT, edge computing, etc.—are integrated rather than used as standalone solutions. To better understand the usage of these technologies in the creation of DT, the top nine software programs that academic papers most frequently mentioned when discussing the creation of DTs is provided in Figure 28. Overall, the most popular software and user interfaces for DT applications include CAD (Computer-Aided Design), MATLAB, and Tecnomatix. While most publications mentioned a particular simulation software by name, they usually failed to mention the specific CAD software they employed; thus, we are grouping all the mentions of CAD into a single category. It is important to note that 33% of the publications reviewed in this study made no indication of the software that was employed. Additionally, forty-five other software packages were referenced in the articles reviewed herein, but none of them were mentioned more than once or twice.

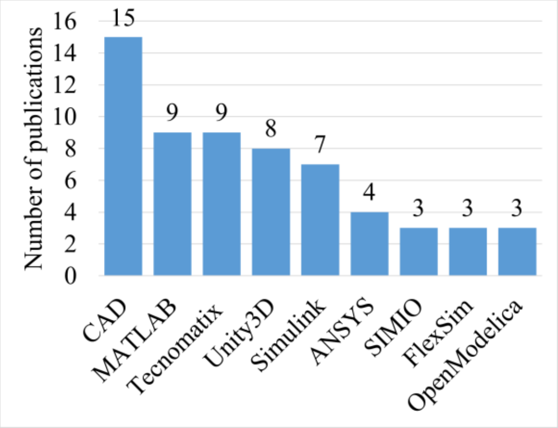


Figure 28 Top Nine Supportive Technologies for Digital Twin.

5.5 Observations and Recommendations

Two main barriers were identified that limit a DT application from reaching a higher level of capability. First, this research has shown that there is still a disconnect between the DT application and concept, as people continue to claim that some traditional simulation models are DTs. Second, those that build a DT consistent with the most popular definitions only make use of a portion of its capabilities. In this section, these issues are presented, and recommendations are provided for how to address them.

Some of the DT applications found in our analysis have been reporting their models as DTs even though they are essentially simulation models. Even though some authors claim to use real-time data connections, in their case studies or application examples, they actually use historical data. This illustrates the challenges related to implementing real-time data transfer. This study's findings reveal that there is still a disconnect between the theory and practice of DT and future work requires effort in filling this gap. This gap will be reduced if there is an agreement on a single DT definition, even though several academics have already identified key DT components and attributes. Calling a simulation model with no real-time synchronization between the physical and virtual environment a DT might be accurate according to some definitions and incorrect according to others. It is crucial for a technology to have a unified definition so that it can be distinguished from other technologies, and this has not been the case for DT.

The existing publishing culture concerning information data sharing also plays a role in this discrepancy between theory and practice of DT. Many DT-related papers are vague about implementation details and do not offer details that could be used as a guide by other researchers. As a result of the lack of sufficient detail in the models presented in papers, the existing work cannot be replicated or improved upon by other researchers, which may cause redundant research and it limits the topic from being widely used. It is challenging to track the implementation process and determine how successful the current DT implementations are without the authors providing in-depth information about the implementation details. While such details may be outside the scope of traditional academic publications, they could be included as online supplements. Changing the publishing culture to encourage more sharing of implementation details is necessary to form a systematic architecture of DT research.

Industrial applications of DT face a hurdle in collecting high-quality data despite it being a requirement, as gathering this data may depend on a variety of factors including the availability of resources to store high-dimensional data and processing capabilities to handle it. This becomes even more challenging to do if it has to happen in real time. Although the current literature may achieve real-time connectivity between physical and virtual spaces, there are few works that implement information flows across the DT's entire lifecycle. Most of the researchers have been attempting to provide methods of tying these real-time data to a DT; however, their work only reaches as far as the R3 level. There hasn't been a research paper where a DT achieved all the 4 levels of capability. Therefore, research on self-evolution and autonomy of the DT is needed. This would enable DT to learn and adapt in real-time.

The SLR reveals that most organizations are not prepared to implement DTs. Furthermore, DT’s complexity varies based on the application sector, current technologies available, and the business objective. Sometimes, the DT can be straightforward and practical, with well-defined parameters and expectations. Other times, it will be an integrated and complicated model requiring more research, specialist knowledge, funding, cost analysis, and upkeep. Adding the complexity of a DT unnecessarily, where a traditional simulation is perfectly capable might be an inherent risk, which can cause issues related to cost, security, and privacy. Inevitably, the creation and continuing maintenance costs for DTs at various levels of complexity will vary, making business cases and return on investment analyses more difficult. Each technology requires investments, therefore choosing the right one to utilize is critical for maximizing its efficiency.

Many of the papers reviewed suggest that industry already employs a wide range of software for routine activities and combining the current software with the DTs software is a challenging task. The data formats and the software tools used by various organizations are often different and these programs frequently do not work together, which causes delays in the adoption of DT. In addition, businesses are hesitant to allow a complete disclosure of data because there is no established protocol. Since a leak of real-time data can be extremely dangerous for a business, a binding legal regulation should be developed across businesses with specified standards for data sharing to have a successful DT operation. Also, platforms that integrate diverse data sources should be researched and blockchain technology should be explored to build DTs to safeguard intellectual property. This allows data to be shared among several parties while still being connected, accessible, and un-editable. Therefore, research on a high-fidelity connection between IoT devices is needed for accurate and timely flow of information.

Another issue that was mentioned frequently in papers studied in this SLR is that as organizations add smarter assets and address more complicated DTs in operations, specific software skills will need to be added to the operations teams. When a project has a short lifespan, a DT can be costly and might not be an investment that businesses are ready to undertake. Similar to other technologies, DT also requires constant updates to reflect advancements in the fields of the technologies that it employs, such as IoT and ML. These technologies provide a DT with the capability of updating itself with the incoming real-time data. Because DT depends on rapidly evolving technologies, this investment in DT needs to be ongoing, which results in higher long-term costs. To prevent needless model complexity and lengthy runtime, it makes sense to model according to the appropriate granularity. Detailed research on this topic is necessary to aid the implementation of the Autonomy property of the DT.

Several authors agree that there are numerous challenges that result from the absence of ML in DT applications currently in use. First, ML optimization problems can be highly complicated, potentially with a number of additional sub problems. Combining these sub problems into one and selecting the right tool to optimize the overall process is a challenge that requires knowledge of both the domain and ML. Second, deep learning implementations within a DT requires computational resources, expertise, and research. Third, a high degree of data quality must be present for the usage of ML to be successful. Data must be accurate to be reliable and useful. Therefore, future research on integrating optimization features for the subcomponents of the DT is needed.

The businesses looking to apply DT should measure the potential business value, limitations, policy, and use. The proposed ways to facilitate the widespread adoption of a fully capable DT are: (1) clearly understand the benefits of implementing a DT before investing in it,

(2) assess if the business is ready to implement IoT technologies that will leverage DTs, (3) examine whether their goals may be achieved using less complex technology, including simulation, (4) develop metrics and indicators to track the development of DT before actually implementing it, and (5) create legal-binding regulations to have an effective DT operation across industries.

5.6 Chapter Summary

This research investigated the connection between simulation and DT, using classification to delineate the differences and similarities in capabilities between them. It also included a discussion of a future agenda for researchers and practitioners. This research assists in clarifying the application of DT by presenting three main contributions:

- A systematic literature review investigated whether the current literature is representative of truly applying DT or if it is using simulation in place of DT.
- The 4S framework and the 4R framework were used to classify the DT applications according to their capabilities.
- There is still a disconnect between the DT concepts and applications, as people continue to construct DTs that are basically simulation models. Those that correctly build a DT only make use of a portion of its capabilities and none of them reached its full capability.

Even though simulation models often use the same type of sensor information as a DT, this information is generated and manipulated as part of the simulation. In other words, a simulation model may replicate in a virtual environment what could happen in the physical environment, but not what is currently happening. Our intention with the classification approach presented in this research was to contribute toward creating a consensus about when a simulation model is a DT and when it is not. The contribution from this chapter and Chapter 4 will be used to provide a way to bridge the gap between simulation and DT, which is presented here.

Chapter 6

Contribution 3 Bridging the Gap between Discrete Event Simulation and Digital Twin

6.1 Introduction

Modern industrial processes are undergoing a transformation regarding how they are planned and conducted, due to data-driven decision support tools like simulation, advanced data analytics, and AI [6]. The concept of DTs can be viewed as a logical progression from conventional simulation modeling when combined with improved data accessibility and connectivity [249]. Both simulation models and DTs have the capability of facilitating the understanding, monitoring, and experimentation on complex physical systems [13,37]. A further capability that DTs present is that they can employ the knowledge obtained from simulations to provide feedback to the physical systems, which can be used to optimize some end-user provided parameters [249]. While a DES model can be an excellent tool for accurately simulating reality in a manufacturing system [9], a DT integrates methods and algorithms from big data analytics and AI, that can faithfully represent what is happening in real-time and provide quick insights on the system that is being modeled [250].

From the technological standpoint, it might be reasonable to consider creative combinations of the current data-driven techniques for using DES. Particularly, approaches from data analytics that aggregate, select, and analyze pre-simulation data may result in successful implementations of DT in the manufacturing practice [251]. Most of the manufacturing applications of simulation in the literature are DES models and DTs of manufacturing systems are expected to continue to use DES models [60,61,250].

This chapter aims to contribute to the field of simulation and modeling by demonstrating the potential for bridging the gap between DES and DT and exploring the extent to which this can be achieved using the available software tools. To achieve these goals, this work will present a case study that documents the process of transforming a fully capable DES (according to the 4S framework) into a DT (according to the 4R framework). Through a case study, this research identifies the key steps involved in this transformation, documents the challenges encountered, and presents recommendations for future studies in this area. By providing a real-world example of how DES can be transformed into a DT, this research aims to contribute to the advancement of simulation and modeling and offer insights into this process and its outcomes.

Note that the approach in this chapter is not to develop a general methodology based on theory and then apply it to the case study. On the contrary, this work follows an inductive approach, whereby the case study is performed first, the process is documented and analyzed, and finally a general framework is inferred and constructed based on the experiences and lessons from the case study.

This chapter is structured as follows: In section 6.2, a methodological approach to this research is presented. Section 6.3 presents a case study of a fully capable DES model. In section 6.4, the empirical process of transforming a DES to a DT is presented. Section 6.5 presents considerations and drawbacks regarding the transformation process. Section 6.6 contains a generalized approach for this transformation. Finally, section 6.7 contains a summary of the contributions of this chapter.

6.2 Methodology

The methodology employed in this research aims to overcome the limitations of current simulation technology and enable the realization of comprehensive and close-to-reality DTs. The main objective is to provide a comprehensive case study that documents the entire journey of transforming a fully capable DES into a DT and to draw lessons from this experience. To realize this objective, the research will first create a fully capable DES that adheres to the levels of capability outlined in the 4S framework (section 4.2). Subsequently, the principles of the 4R framework will be applied to this DES model to achieve as much capability of a DT as possible, documenting and analyzing the steps taken during this transformation. The findings will encompass the lessons learned and the identified limitations. These insights will then be synthesized into a framework that outlines the progression from the 4S framework to the 4R framework.

Through the implementation of this case study, the research aims to provide a practical demonstration of bridging the gap between DES and DT, while also drawing insight to develop a generalized approach that can be applied to other cases. This research focuses on experimentally achieving as much capability of a DT as possible; however, due to technological and bureaucratic constraints, only the Representation (R1) level and partially the Replication (R2) level are achieved. These levels encompass the process of capturing and representing the essential characteristics of the physical system, as well as the replication of its behavior in the digital environment. It should be noted that the levels of Reality (R3) and Relational (R4) will be addressed in future studies.

6.3 DES: Pallet Elevator in an Automotive Engine Manufacturing Plant

In this section, a fully capable DES that adheres to the levels of capability outlined in the 4S framework is presented. This DES model will serve as a foundation for the subsequent transformation into a DT. The case study presented in this chapter is conducted within a different section of the same automotive engine manufacturing plant that was studied in Chapter 4. The purpose of this chapter is to describe the details of that DES, including its purpose, scope, and key components.

The automotive engine block manufacturing plant is divided into three sections: die-casting, machining, and engine assembly. Within the engine assembly section, there are four distinct lines: piston line, head line, block line, and main line. The focus of this study is specifically on the pallet elevator area, which serves as the connection between the head line and the main line. The engine assembly section was chosen for this study due to its elevated level of automation, sensors, and the presence of a real-time data connection. This advantageous combination enables the effective utilization of the data connection to construct a DT through simulation. The pallet elevator area is equipped with sensors and controlled using Programmable Logic Controllers (PLCs), allowing for the monitoring and analysis of the decision-making processes. This aspect is highly valuable for the translation of these processes into a DT, especially when compared to the block machining area (considered in the DES presented in Chapter 4), which lacks IoT functionality.

6.3.1 Modeling (S1)

A conceptual model of the pallet elevator area is depicted in Figure 29. The pallet flow within this system operates as follows: when the head line reaches its end, a full pallet containing two engine heads is transported to the pallet elevator. The pallet elevator then carries the pallet to an overhead conveyor, called the “Head Buffer.” This conveyor will take the pallet to the “Head Install” machine, where the engine heads are removed from the pallet and delivered to the Main Line. At this point, the pallet is empty and needs to return to the pallet elevator using the “Head Return” conveyor. Each pallet is then transported down, using the elevator, to the next machine in line, labeled “Head Load B,” where engine heads are loaded back onto the pallet. The full pallet continues its journey through a conveyor, passing through various machines within the head line until it reaches the pallet elevator once again. This conveyor system forms a closed loop. It is important to note that there is only one pallet elevator responsible for the task of moving pallets up or down, one at a time. (Certain technical details of this elevator area are being omitted to preserve confidentiality and safeguard the interests of the industry partner.)

The specific area of interest for developing a DES and transforming it into a DT is visually marked in a black square in Figure 29. This area begins when a pallet is delivered to the elevator from the head line, continues as the pallet goes through the Head Install machine, and concludes when a pallet departs from the Head Load B machine.

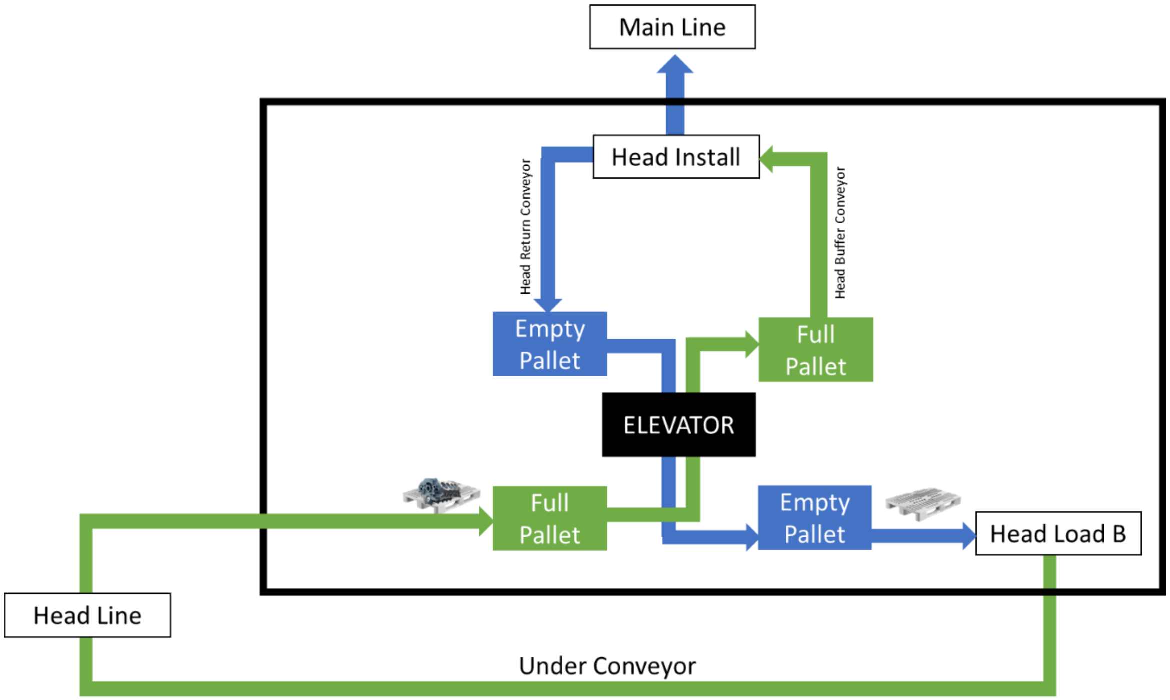


Figure 29 Pallet Elevator Conceptual Model.

The operational schedule of this system follows a two-shift system with a 30-minute lunch break and a 10-minute break within each shift. Additionally, there is a time interval of one hour and thirty minutes between the completion of one shift and the start of the next. It should be noted, however, that the breaks do not apply to the elevator and the Head Load B machine due to automated operation of the equipment. Also, historical data was collected to be fitted into probability distributions for each machine, such as downtime, processing time, and reliability logic. Note that all the assumptions listed here can be lifted if live data is available, as that data would capture these situations.

The elevator system in the engine assembly area has the capability to travel with or without a pallet. The managers of this area are interested in exploring whether there is a more efficient logic for the elevator that could reduce the occurrence of “travel without a pallet” instances or provide overall benefits to the assembly line. The current logic for the elevator has been in place for a significant period of time without any modifications, prompting the managers to question whether there might be an improved approach worth considering.

This pallet elevator is controlled by a PLC logic and the decision-making process is based on pallet presence sensors in the four corners of the elevator. The logic allows the elevator to assess the situation and make appropriate decisions for pallet movement, ensuring efficient operation within the assembly line. The elevator logic has been decoded into the following decisions points:

- When a pallet arrives at the elevator, whether it is a full pallet from the head line or an empty pallet from the overhead “Head Return” conveyor, the elevator determines its current position (up or down).
- It checks the presence of a pallet in the required corner’s destination to make the decision. This step is crucial as the elevator will only transport a pallet to its destination if there is sufficient space available at the destination to accommodate a pallet.
- The possible decisions at this point are: (1) Do not release the pallet into the elevator, and the elevator remains in its current position. (2) Release the pallet into the elevator and raise or lower it accordingly. (3) Raise or lower the elevator without a pallet.

Once the system’s current state was understood, it became possible to portray the elements of the conceptual model in a virtual environment. To ensure consistency, the decision was made to continue utilizing Simio as the simulation software for creating the virtual model, as it had been employed in the previous case study demonstration. The current state of the pallet elevator area was conceptually defined as the “Baseline Model” (BL). A snapshot of the simulation model created using Simio is displayed in Figure 30. This system consists of three sources, three servers, one vehicle, and one sink. Source1 supplies the arrival rate of pallets into the system, while Source2 and Source3 add pallets to the "Head Buffer" and “Head Return” conveyor, respectively, at the beginning of the run. The purpose of having Source2 and Source3 provide pallets at the start of the run is to depict the system's usual starting condition at the beginning of the day. Two of the servers represent the machines Head Install and Head Load B, while an additional server named Server1 is included solely to enable the incorporation of a work schedule for Source1. This is necessary because the Simio software used in the system does not directly support assigning work schedules to sources. To simulate the functioning and movement of pallets vertically, both upwards and downwards, a vehicle is employed as a representation of the elevator. The decision to use a vehicle is motivated by its flexibility in manipulating operational logic and transporting pallets. Also, multiple conveyors are employed to establish connections between the source, servers, and sink within this system.

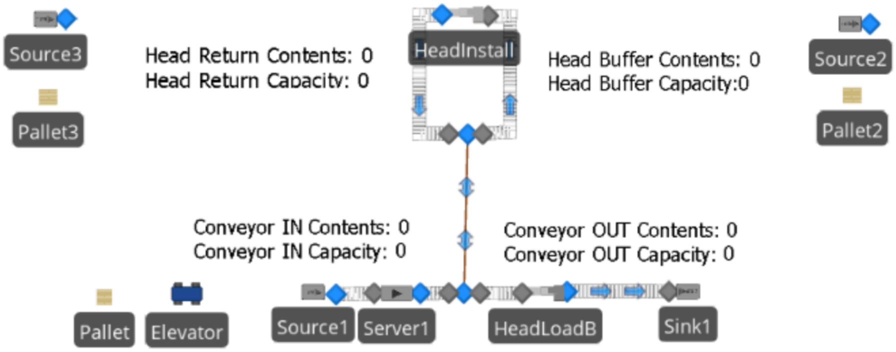


Figure 30 Snapshot of the DES model of the Pallet Elevator Area.

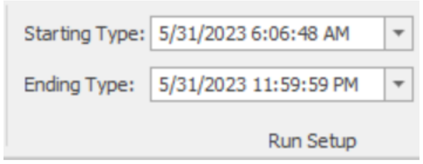
6.3.2 Analyzing (S2)

In the second level of the 4S framework, Analyzing, an input analysis is performed and the historical data that was collected in the previous phase is connected to the computer model. The model is then verified and validated, and analysis of the system in its current state is performed.

6.3.2.1 Input Analysis

In this case study, historical data was gathered and incorporated into the model elements to determine the arrival rate of pallets and reliability logic for Head Install and Head Load B machines. Other data utilized in the simulation model included the conveyor’s speed and length, as well as the processing times of the machines.

To determine the input arrival rate for the simulation model, information was gathered throughout a day, encompassing two working shifts. This collected data comprises timestamps indicating when a pallet is prepared for loading onto the elevator. These timestamps are employed as an arrival data table for the simulation model, to reflect the genuine quantity of pallets entering the system throughout the day. This decision to utilize the timestamps as the arrival table, rather than employing random distributions, aims to ensure a realistic depiction of pallet arrivals in the system. The simulation run is initiated on the date of 05/31/2023, starting at 6:06:48 am and concluding at 11:59:59 pm (Figure 31). The decision to gather data that corresponds one day instead of collecting data for a month, for instance, is rooted in the objective of understanding the system’s transient behavior during a typical day rather than exploring potential long-term behaviors. Furthermore, the need for a warm-up period is eliminated due to a consistent initial condition that remains the same at the start of each shift. This initial condition entails a full head buffer conveyor and an empty head return conveyor. This is achieved because the elevator remains operational until all the pallets in the system have been moved. Consequently, the elevator continues functioning even after a shift ends, ensuring that the head return buffer is completely emptied. As a result, the start of the next day's shift always begins with a full head buffer conveyor and an empty head return conveyor. This reset that the system experiences each day led to the decision to run many replications of a single day, rather than run many days in a row in each replication, as each day is fairly independent of the ones before or after it.



Starting Type: 5/31/2023 6:06:48 AM

Ending Type: 5/31/2023 11:59:59 PM

Figure 31 Run Setup for the Simulation Model.

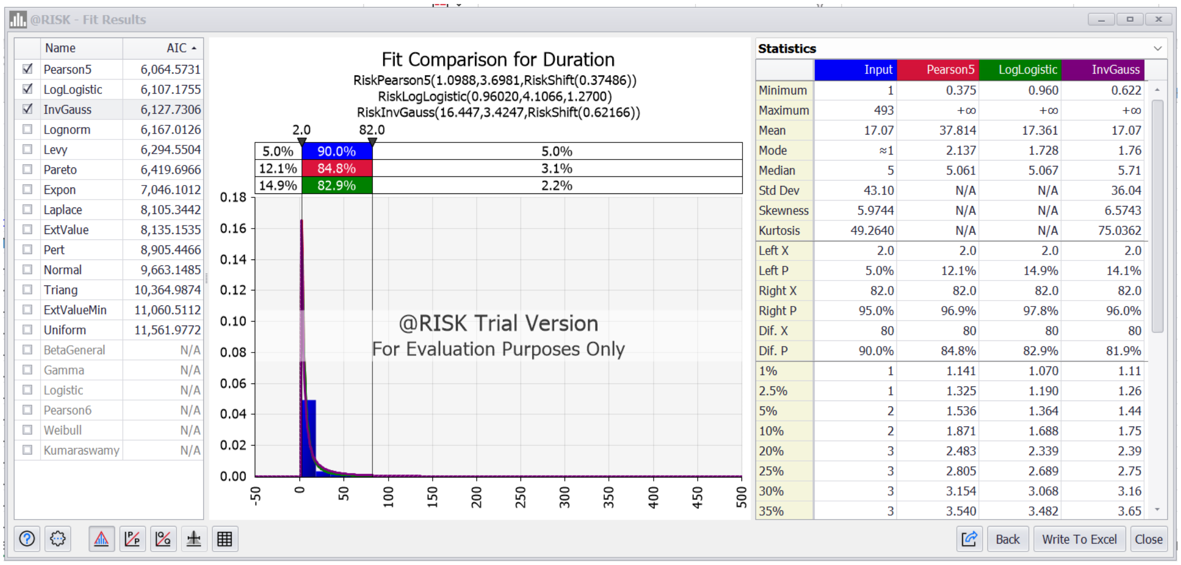
To evaluate the reliability of the Head Install and Head Load B servers, data regarding downtime for these machines was collected for the same one-day period. This data was analyzed using @Risk, an add-in tool for excel, to identify the most appropriate distribution that effectively represents the collected data for the time between failures and the time required for repairs [252]. The time between failures represents the duration during which the machines remain operational without experiencing any failures or breakdowns. The time to repair indicates the amount of time needed to fix a machine after it has encountered a failure or breakdown. Table 40 exhibits the distributions employed for the input analysis of this model, including the arrival time rate, time between failures, and time to repair.

Table 40 Input Analysis Data Used in the Simulation Model.

Process Data Input in DES		
Source	Arrival Rate	Arrival Table with Timestamps
Head Install	Uptime between failures	Random.LogLogistic(9.7235,38.828) seconds
	Time to repair	Random.LogLogistic(1.2700 ,4.1066) seconds
Head Load B	Uptime between failures	Random.Exponential(153.14) seconds
	Time to repair	Random.Lognormal(1.1147,0.4187) seconds

A comprehensive evaluation, encompassing both quantitative and qualitative aspects, was conducted using @Risk to determine the most suitable distribution for each machine based on the provided input data. The distributions were ranked using the AIC score, and a visual comparison was made between the probability density function of the fitted distribution and the input data. For each fitted distribution, essential statistics such as mean, standard deviation, mode, etc., were calculated and reported. These statistics can be easily compared to the corresponding statistics of the input data. Figure 32 presents the top four distributions that best fit the time to repair of the Head Install machine. Although the AIC classification suggests that the Pearson5 distribution is the best fit, it is not available in Simio. Consequently, the LogLogistic distribution was chosen as the representative distribution for this data.

Similarly, the process was repeated to identify the best fit for the uptime between failures of the Head Install Machine. Figure 33 illustrates the top four distributions that demonstrated the closest fit to the data. Among them, the LogLogistic distribution emerged as the best-fitting distribution for this particular case.



	Name	AIC
☑	Pearson5	6,064.5731
☑	LogLogistic	6,107.1755
☑	InvGauss	6,127.7306
☐	Lognorm	6,167.0126
☐	Levy	6,294.5504
☐	Pareto	6,419.6966
☐	Expon	7,046.1012
☐	Laplace	8,105.3442
☐	ExtValue	8,135.1535
☐	Pert	8,905.4466
☐	Normal	9,663.1485
☐	Triang	10,364.9874
☐	ExtValueMin	11,060.5112
☐	Uniform	11,561.9772
☐	BetaGeneral	N/A
☐	Gamma	N/A
☐	Logistic	N/A
☐	Pearson6	N/A
☐	Weibull	N/A
☐	Kumaraswamy	N/A

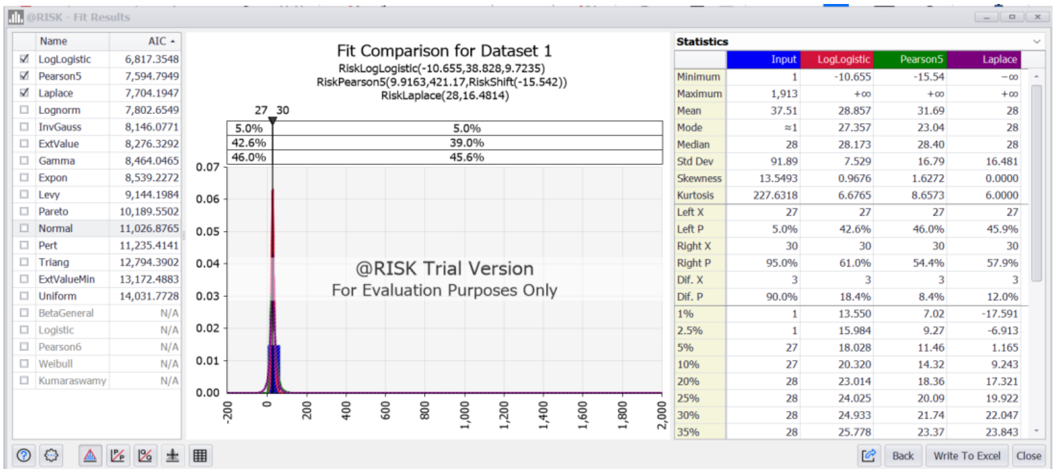
RiskPearson5(1.0988,3.6981,RiskShift(0.37486))
RiskLogLogistic(0.96020,4.1066,1.2700)
RiskInvGauss(16.447,3.4247,RiskShift(0.62166))

5.0%	90.0%	5.0%
12.1%	84.8%	3.1%
14.9%	82.9%	2.2%

Statistics

	Input	Pearson5	LogLogistic	InvGauss
Minimum	1	0.375	0.960	0.622
Maximum	493	+∞	+∞	+∞
Mean	17.07	37.814	17.361	17.07
Mode	≈1	2.137	1.728	1.76
Median	5	5.061	5.067	5.71
Std Dev	43.10	N/A	N/A	36.04
Skewness	5.9744	N/A	N/A	6.5743
Kurtosis	49.2640	N/A	N/A	75.0362
Left X	2.0	2.0	2.0	2.0
Left P	5.0%	12.1%	14.9%	14.1%
Right X	82.0	82.0	82.0	82.0
Right P	95.0%	96.9%	97.8%	96.0%
Dif. X	80	80	80	80
Dif. P	90.0%	84.8%	82.9%	81.9%
1%	1	1.141	1.070	1.11
2.5%	1	1.325	1.190	1.26
5%	2	1.536	1.364	1.44
10%	2	1.871	1.688	1.75
20%	3	2.483	2.339	2.39
25%	3	2.805	2.689	2.75
30%	3	3.154	3.068	3.16
35%	3	3.540	3.482	3.65

Figure 32 Fitting Distribution to Head Install’s Time to Repair.



	Name	AIC
☑	LogLogistic	6,817.3548
☑	Pearson5	7,594.7949
☑	Laplace	7,704.1947
☐	Lognorm	7,802.6549
☐	InvGauss	8,146.0771
☐	ExtValue	8,276.3292
☐	Gamma	8,464.0465
☐	Expon	8,539.2272
☐	Levy	9,144.1984
☐	Pareto	10,189.5502
☐	Normal	11,026.8765
☐	Pert	11,235.4141
☐	Triang	12,794.3902
☐	ExtValueMin	13,172.4883
☐	Uniform	14,031.7728
☐	BetaGeneral	N/A
☐	Logistic	N/A
☐	Pearson6	N/A
☐	Weibull	N/A
☐	Kumaraswamy	N/A

	Input	LogLogistic	Pearson5	Laplace
Minimum	1	-10.655	-15.54	−∞
Maximum	1,913	+∞	+∞	+∞
Mean	37.51	28.857	31.69	28
Mode	≈1	27.357	23.04	28
Median	28	28.173	28.40	28
Std Dev	91.89	7.529	16.79	16.481
Skewness	13.5493	0.9676	1.6272	0.0000
Kurtosis	227.6318	6.6765	8.6573	6.0000
Left X	27	27	27	27
Left P	5.0%	42.6%	46.0%	45.9%
Right X	30	30	30	30
Right P	95.0%	61.0%	54.4%	57.9%
Dif. X	3	3	3	3
Dif. P	90.0%	18.4%	8.4%	12.0%
1%	1	13.550	7.02	-17.591
2.5%	1	15.984	9.27	-6.913
5%	27	18.028	11.46	1.165
10%	27	20.320	14.32	9.243
20%	28	23.014	18.36	17.321
25%	28	24.025	20.09	19.922
30%	28	24.933	21.74	22.047
35%	28	25.778	23.37	23.843

Figure 33 Fitting Distribution to Head Install's Uptime Between Failures.

A similar procedure was employed to determine the most suitable distributions for the Head Load B machine. Unlike the Head Install machine, downtimes for this machine occur less frequently. It was found that the exponential distribution provides the best fit for the uptime between failures, while the Lognormal distribution is the most appropriate for modeling the duration. In the next section, these fitted distributions for both machines will be validated to ensure that they accurately represent the system.

Additionally, other complementary data, such as processing times and conveyor information, were collected and are presented in Table 41. These physical data were acquired in collaboration with the engineers who are actively involved in the operational environment. They had access to the relevant sources and systems, such as the CAD files of the conveyors, which facilitated the collection of the necessary data.

Table 41 Physical Data Used in the Simulation Model.

Physical Data Input in DES		
Head Install	Processing time	28.1 seconds
Head Load B	Processing time	21.7 seconds
Conveyor	Speed Length Capacity	21 meters per minute Several conveyors with different lengths Several capacities
Elevator	Cycle time	13 seconds

The automotive industry partner does not collect time stamps for processing times; instead, they initially provided a constant value, which is supposed to represent the actual cycle time. However, when using these constant values, the validation of the model was not successful. In consultation with the engineers in the area, we concluded that the issue was that the cycle times are not really constant. As a result, it was decided to introduce variability in the processing times for both the Head Install and Head Load B machines. The engineers from the area agreed to use the triangular distribution for modeling the processing times of the Head Install and Head Load B machines. The triangular distribution is a suitable choice when there is limited knowledge about the exact distribution shape, but some information about the minimum, maximum, and most likely values is available. In this case, even though quantitative data was unavailable, the engineers were able to provide their best estimates for the minimum, maximum and the mode of the cycle times. For the Head Install machine, the triangular distribution (28, 31, 35) was chosen. This means that the minimum processing time is 28, the maximum processing time is 35, and the most likely or mode value is 31. Similarly, for the Head Load B machine, the triangular distribution (21.7, 23, 30) was selected. The validation of the model was successful with this approach.

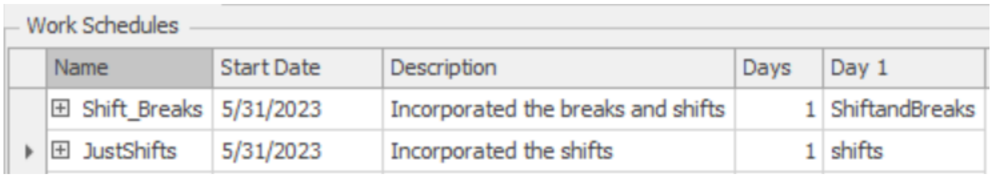
6.3.2.2 Verification and Validation of the DES

Verification was conducted as an ongoing process as the simulation model was developed. The company operates its production during weekdays only, resulting in a total of five production days per week. As the company doesn't run 24 hours, 7 days per week, work schedules were implemented for each server, which determine when they can run. These work schedules consist of two shifts lasting eight hours each, with designated break times, as shown in Table 42.

Table 42 Work Schedules for Each Shift.

1st Shift	**Type**	**2nd Shift**
6:00 am	Start	4:00 pm
8:30 – 8:40 am	10 min break	6:30 – 6:40 pm
10:30 – 11:00 am	30 min lunch break	8:30 – 9:00 pm
1:00 –1:10 pm	10 min break	11:00 – 11:10 pm
2:30 pm	End	12:30 am

As mentioned earlier, there are two machines in this system: Head Install and Head Load B. Since Head Load B is automated, it does not take breaks in the shifts schedule. Therefore, two different types of work schedules were implemented, as shown in Figure 34. Both Head Load B machine and the elevator follow the JustShifts schedule, whereas the Head Install machine and the server follow the Shift_Breaks schedule.


Work Schedules

Name	Start Date	Description	Days	Day 1
⊞ Shift_Breaks	5/31/2023	Incorporated the breaks and shifts	1	ShiftandBreaks
⊞ JustShifts	5/31/2023	Incorporated the shifts	1	shifts

Figure 34 Types of Work Schedules in the Simulation Model.

During non-production periods, the servers are switched to an off-shift mode and stop running. When servers are in an off-shift mode, they appear white in color (Figure 35). This visual cue serves as a clear indicator that the servers are not currently operational and are following the designated off-shift schedule. By employing this off-shift mode and the corresponding white color appearance, the work schedules were verified and confirmed that they are accurately implemented, allowing for convenient identification of the servers' operational state at any given time.

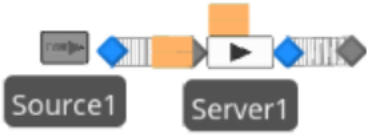


Figure 35 Off-shift Mode Verification.

To monitor and track various metrics within the simulation, status labels were implemented. These labels were designed to provide information on the number of pallets present in the conveyors surrounding the elevator, as well as the capacity of these conveyors. Figure 36 illustrates the presence of these three status labels in the simulation. At the beginning of the simulation run, the system accurately generated the appropriate entities. This ensured that the simulation closely replicated the real-world scenario.

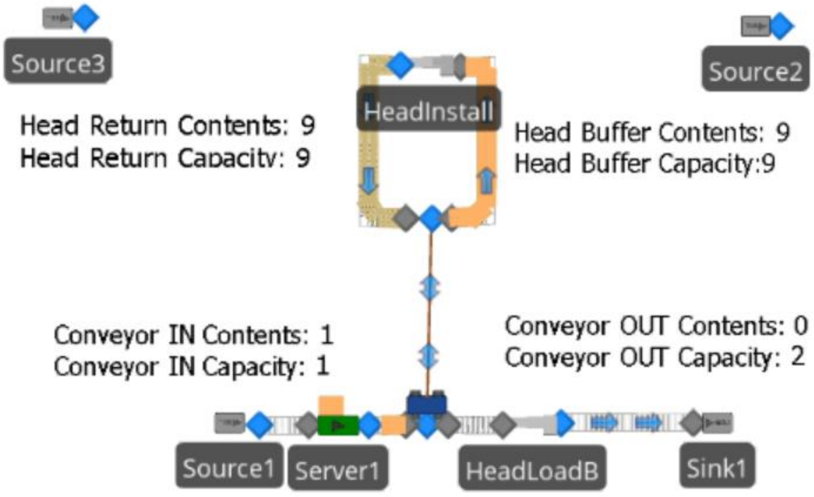


Figure 36 Verification Using Status-labels.

Throughout the construction of the simulation, the utilization of Trace Windows and Watch Windows was frequent, serving to enhance the intricacy of the model. These tools are valuable for verification and debugging of the simulation model within Simio. The Trace Window presents a visual depiction of the sequential logic carried out by objects and tokens within the model, which allows for monitoring the flow of operations and detecting possible problems or mistakes [39]. On the other hand, the Watch Window offers instantaneous information by presenting the present values of states, properties, functions, and elements during an interactive run [39].

Similarly to the case study discussed in section 4.3, a quantitative process to determine the appropriate number of replications was performed. Table 43 presents the half-width results obtained by running 10, 30, and 50 replications. The findings clearly demonstrate that the improvements achieved by using more than 30 replications are relatively insignificant compared to the gains observed before reaching 30 replications. Consequently, 30 replications was selected for this particular case study. This number represents a midpoint between 10 and 50 replications, and the area managers have agreed that the results obtained from these 30 replications are sufficient to represent the system accurately.

Table 43 Number of Replications Experiment.

Head Install Failures	**10 Replications Simulation Data ± Half-Width**	**30 Replications Simulation Data ± Half-Width**	**50 Replications Simulation Data ± Half-Width**
Average (sec)	13.50 ± 1.33	14.02 ± 1.12	14.04 ± 1.06
Occurrences	962.1 ± 17.00	950.9 ± 16.15	950.4 ± 15.77
Total Time (sec)	12957 ± 1095.9	13221 ± 798.49	13111 ± 681.24
Total Time (%)	20.12% ± 1.70	20.53% ± 1.24	20.36% ± 1.05

To validate the fitted distributions for the Head Install machine, the simulation results from the 30 replications column were compared to the real data (Table 44). The average duration of failures obtained from the simulation showed only a 9% difference when compared to the real data, with a 2% relative error when assessing the occurrences of failures. The total time of failure and the corresponding percentage demonstrated a 6% relative error. The managers responsible for this area agreed that these differences were sufficiently representative of the system.

Table 44 Head Install Validation.

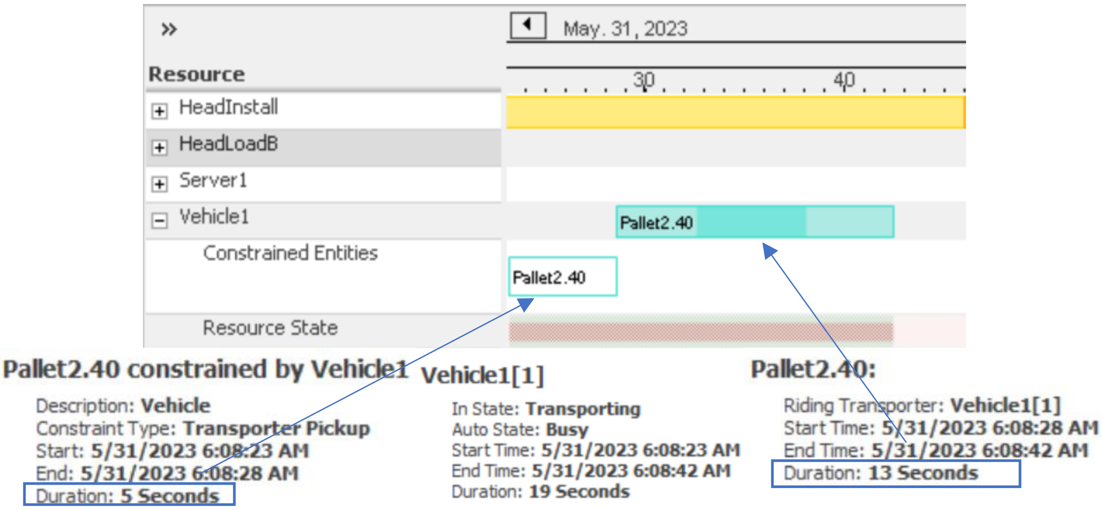
Head Install Failures	**Real Data**	**30 Replications Simulation Data ± Half-Width**	**Relative Error**
Average (sec)	15.40	14.02 ± 1.12	9%
Occurrences	932	950.9 ± 16.15	2%
Total Time (sec)	14208	13221 ± 798.49	6%
Total Time (%)	22%	20.53% ± 1.24	6%

The elevator cycle time was subjected to validation as well. Table 45 presents the results obtained from comparing the real data with the simulation data (Figure 37). According to the information provided, the duration of the elevator ride, whether going up or down, is 4.9 seconds. However, the complete cycle time, which includes the times for pallet load and unload, totals 13 seconds. Upon reviewing these results, the managers responsible for the area confirmed that they are representative of the actual elevator performance.

Table 45 Elevator Time Validation.

Elevator	**Real Data**	**Simulation**	**Relative Error**
Elevator Ride Up or Down (sec)	4.9	5	2%
Elevator Cycle Time (sec)	13	13	0%

A portion of the Gantt chart, shown in Figure 37, serves as evidence to confirm the proper functioning of the elevator. Specifically, it illustrates the movement of a particular pallet, encompassing both the loading and unloading durations, which amounts to 13 seconds. However, the total transportation time is recorded as 19 seconds, as it includes an additional 5 seconds during which the elevator was in motion, solely to retrieve the pallet.



Pallet2.40 constrained by Vehicle1
Description: **Vehicle**
Constraint Type: **Transporter Pickup**
Start: **5/31/2023 6:08:23 AM**
End: **5/31/2023 6:08:28 AM**
Duration: **5 Seconds**

Vehicle1[1]
In State: **Transporting**
Auto State: **Busy**
Start Time: **5/31/2023 6:08:23 AM**
End Time: **5/31/2023 6:08:42 AM**
Duration: **19 Seconds**

Pallet2.40:
Riding Transporter: **Vehicle1[1]**
Start Time: **5/31/2023 6:08:28 AM**
End Time: **5/31/2023 6:08:42 AM**
Duration: **13 Seconds**

Figure 37 Simulation Results of the Elevator Times.

To complete the model validation process, the starved time for the Head Install machine was recorded and compared to the actual data reflecting a typical day of production. The results of this comparison are presented in Table 46, indicating a relative error of only 3.1% between the simulation data and the real data. The real value falls within the confidence interval, suggesting that the simulated starved time for this machine is representative enough to accurately capture its behavior, according to the assessment of the managers of this area.

Table 46 Head Install Starved Time.

Head Install Starved time	**Real Data**	**Simulation Data ± Half-Width**	**Relative Error**
Total Time (sec)	2096	2160 ± 315.8	3.1%

6.3.2.3 Current State Analysis

After collecting the data and verifying and validating the model, the metrics used to analyze the system in further phases are determined. In this case study, the model tracks the following measures of performance:

a. Head Install starved time: This is the primary measure of performance of this study. The objective is to investigate if modifying the elevator's operational logic could minimize the time during which Head Install experiences starvation, which refers to the period during which a machine is unable to operate due to a lack of necessary inputs or resources. As previously stated, this particular machine acts as a link between two production lines, and it is crucial to minimize any interruptions in the flow in order to meet the daily production targets.

b. Elevator movement time with pallet: This represents the total time that the elevator is engaged in transporting pallets. It encompasses the entire timeframe required for tasks such as loading and unloading the pallet onto the elevator, as well as the actual transportation process.

c. Elevator movement time without pallet: At certain times, the elevator in this area has to move without carrying a pallet. This occurs when the elevator is stationed in a location separate from where the assigned pallet needs to be transported. In such cases, the elevator will travel empty to reach the designated pallet and then proceed with its transportation.

d. Elevator idle time: This refers to the period when the elevator is not engaged in any tasks and is waiting for a pallet to arrive in order to initiate transportation. During this idle time, the elevator remains inactive until a pallet becomes available for transport.

e. Head Load B starved time: this refers to the period during which Head Load B machine is unable to operate due to a lack of necessary inputs or resources.

The results for the measures of performance on the current state of the baseline model are presented in Table 47.

Table 47 Results of Measures of Performance of Baseline Model.

Measures of Performance	**Baseline Model**
Head Install Starved Time	2160 seconds (3.35%)
Elevator Movement With Pallet	32682.53 seconds (50.74%)
Elevator Movement Without Pallet	1487.67 seconds (2.31%)
Elevator Idle Time	30220.8 seconds (46.93%)
Head Load B Starved Time	34572.5 seconds (53.7%)

6.3.3 Predicting (S3)

At this level, the simulation model has the capability to anticipate and forecast the performance of the physical system based on specific inputs. To incorporate this level into our simulation model, a what-if scenario was created by modifying both the inputs and structure of the model. The objective was to examine the impact on the selected measures of performance when the elevator logic was altered.

Alternative Elevator Logic (AEL): This new logic was derived through extensive discussions with the engineers working in the area, who contributed their expertise to select these modifications. The only difference in the elevator logic, when comparing to the Baseline Model, is that one extra decision is made. In the AEL scenario, if the elevator is in a low position and the destination conveyor (head buffer conveyor) contains five or more pallets, while the other conveyor (return conveyor) has five or less pallets, the decision is made to get two pallets consecutively from the return conveyor without delivering pallets to the head buffer, carrying one pallet at a time. The number five was chosen by the managers in this area, who consider it to be a safe threshold for making this decision. This decision is made to explore the potential benefits of intentionally moving the elevator without pallets more frequently, with the aim of creating more space in the return buffer. Choosing a threshold that is too low could result in unnecessary disruptions to the regular flow of pallets, potentially leading to delays or inefficiencies. On the other hand, setting a threshold that is too high might limit the opportunity to exploit the benefits of intentionally moving the elevator without pallets more frequently.

Table 48 presents the results obtained from the Alternative Elevator Logic model, which implemented new elevator logic, and compares them to the Baseline Model, which utilized the original elevator logic. The results indicate that, overall, the Alternative Elevator Logic model had a negative impact on all the performance measures. While the increase in starved time for Head Install was relatively small, only 0.18% in total, it still represented an increase compared to the Baseline Model. This finding challenges the hypothesis that changing the elevator logic would help minimize the starved time for Head Install, which is interesting as it contradicts the beliefs held by area managers. Also, it is worth noting that the slight increase in elevator movement without pallets did not have a positive effect on the starvation time of Head Install. This suggests that the anticipated benefits of increased elevator movement did not translate into reduced starved time as expected.

Table 48 Comparison of Results from Baseline Model and Alternative Elevator Logic.

Measures of Performance	Baseline Model (BL)	Alternative Elevator Logic (EL)	Difference
Head Install Starved Time	2160 seconds (3.35%)	2273 seconds (3.53%)	+0.18%
Elevator Movement With Pallet	32682.53 seconds (50.74%)	33009.8 seconds (51.26%)	+0.52%
Elevator Movement Without Pallet	1487.67 seconds (2.31%)	1522 seconds (2.36%)	+0.05%
Elevator Idle Time	30220.8 seconds (46.93%)	29859 seconds (46.37%)	-0.56%
Head Load B Starved Time	34572.5 seconds (53.7%)	34191.9 seconds (53.1%)	+0.60%

6.3.4 Prescribing (S4)

In the last level of the 4S framework, Prescribing, a simulation optimization technique is utilized to determine the optimal course of action. Similar to the previous case study described in section 4.3, the simulation optimization approach employed is the OptQuest add-in in Simio.

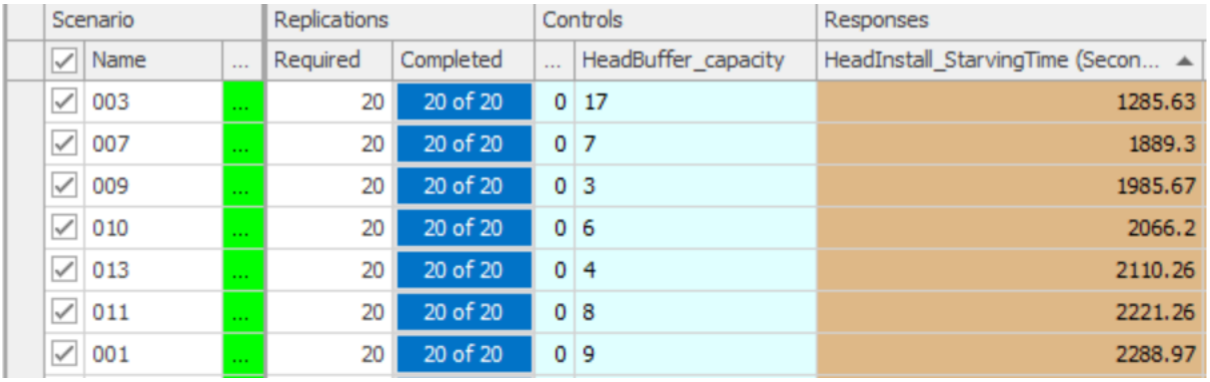
The primary objective of employing OptQuest in this particular context is to analyze the impact of changing the relative position of the Head Install on performance metrics. Specifically, in section 6.3.1, it is mentioned that there are two conveyors surrounding the Head Install: the head buffer (prior to the Head Install) and the head return (after the Head Install). The combined capacity of these conveyors is 18 pallets. The aim is to determine the optimal distribution of this conveyor capacity to yield better results across performance measures.

For conducting this experiment in OptQuest, only the head buffer conveyor was considered as the control variable. The capacity of the head return conveyor was determined by subtracting the capacity of the head buffer conveyor from the total capacity of 18 pallets. Table 49 provides the details of the control variable used in the experiment, specifying the minimum value, maximum value, and increment size. The range for this experiment was set from 1 to 17, excluding the possibility of a buffer with 0 capacity.

Table 49 OptQuest Control Values.

Minimum Value	**Controls**	**Maximum Value**	**Increment Size**
1	Head Buffer Conveyor	17	1

The primary focus is on minimizing the Head Install starved time, which is the key performance metric. After the completion of OptQuest runs, the Head Install starved time column was sorted in ascending order, and the smallest value for each model was selected. An illustration of the OptQuest results in Simio can be observed in Figure 38 .



Scenario			Replications		Controls		Responses
☑	Name	...	Required	Completed	...	HeadBuffer_capacity	HeadInstall_StarvingTime (Secon... ▲
☑	003	...	20	20 of 20	0	17	1285.63
☑	007	...	20	20 of 20	0	7	1889.3
☑	009	...	20	20 of 20	0	3	1985.67
☑	010	...	20	20 of 20	0	6	2066.2
☑	013	...	20	20 of 20	0	4	2110.26
☑	011	...	20	20 of 20	0	8	2221.26
☑	001	...	20	20 of 20	0	9	2288.97

Figure 38 OptQuest Results.

Table 50 provides an overview of the OptQuest results obtained from the Baseline and Alternative Elevator Logic models. The table showcases the optimal control values and their corresponding results for the measures of performance for both models. The optimal control values that yielded lower Head Install starvation time were found to be a head buffer capacity of 17 and a head return capacity of 1 in both the Baseline and Alternative Elevator Logic models. These control settings proved to be effective in minimizing the time during which the Head Install process experienced starvation.

When comparing the results between the Baseline and Alternative Elevator Logic models, it is noteworthy that, in this particular instance, the Alternative Elevator Logic model performed slightly better overall compared to the baseline model. This outcome differs from previous comparisons where only the Alternative Elevator Logic model was applied with the existing 9x9 configuration. These findings suggest that the Alternative Elevator Logic model, with optimized control values, has the potential to enhance the overall system performance and reduce Head Install starvation time more effectively.

Table 50 OptQuest Results for the Baseline and Alternative Elevator Logic Models.

	Control	Responses				
	Head Buffer Capacity	Head Install Starved Time	Elevator With Pallet	Elevator Without Pallet	Elevator Idle Time	Head Load B Starved Time
Baseline	17	1285.6 sec (1.99%)	32146.4 sec (49.92 %)	1609 sec (2.49%)	30635.6 sec (47.57%)	35106.4 sec (54.52%)
Alternative Elevator Logic	17	1073.6 sec (1.66%)	32407.9 sec (50.32%)	1643.7 sec (2.55%)	30339.3 sec (47.11%)	34764.8 sec (53.99%)

By systematically exploring various configurations and sorting the results, OptQuest played a crucial role in effectively identifying optimal solutions. In the discussed case, OptQuest assisted in selecting control parameters that minimized Head Install starved time. This final level of capability demonstrated by the simulation model offers valuable insights for making informed decisions regarding optimal system operation or configuration. Having achieved all four levels of capability within the 4S simulation framework, the model can be regarded as fully capable.

6.4 Process of Transforming the Pallet Elevator DES to a DT

In this section, a comprehensive case study that documents the entire process of transforming a fully capable DES into a DT is presented. The crucial steps involved in this transformation are identified and presented in this section. To reiterate, this research initially attempted to achieve all four levels of capabilities of a DT according to the 4R framework. However, only the initial levels were achieved and presented here: Representation (R1) and Replication (R2). The subsequent levels, Reality (R3) and Relational (R4), are not encompassed due to technological limitations and bureaucratic constraints. Overcoming these limitations is beyond the scope of this study and will be the subject of future research efforts.

6.4.1 Achieving Representation (R1)

To transform a DES to a DT, achieving the first capability level of the 4R framework, which is Representation (R1), several key steps were taken. The process began by establishing the purpose of transforming a pallet elevator DES into a DT. The next step involved mapping the data elements used in the existing DES. This included identifying the different data types used in the system, such as processing times, capacity, interarrival time, and work schedules. To facilitate the collection, storage, and integration of data, a data architecture was formulated. This architecture provided a structured and formal description of the system, highlighting its structure and behaviors. Finally, with the data architecture in place, the data collection process was executed. Below, each of these steps are described in detail.

6.4.1.1 Step 1: Define the purpose of Transforming a DES into a DT

Similarly to the simulation-building process, before creating a DT, the objectives and expectations must be well understood [250]. In this case study, the objective was to create a DT using a DES model within a manufacturing setting. The aim for creating the DES model of the pallet elevator, presented in the previous section, was to verify and assess the consequences of potential modifications to the elevator logic. The goal for creating a DT from the DES was to integrate real-time data from various sources, including sensors and equipment, to facilitate real-time decision-making. As this elevator area is automated and equipped with sensors, the desire was not only to utilize simulation to explore potential scenarios but also to establish a connection between the simulation and real-time data from the production line. This connection would enable the prediction of changes that closely resemble real-life conditions.

When developing a DT, it is crucial to have a deep understanding of the behavior of the physical system [92]. In this case, this understanding has already been established through the development of the DES model. This existing knowledge gained from the DES serves as a foundation for developing the DT, allowing for a comprehensive understanding of the physical system’s functioning, performance characteristics, and operational constraints. The insights and understanding acquired from the DES can guide the development of the DT by providing essential information about the system’s behavior, expected responses, and critical factors affecting its performance [249].

6.4.1.2 Step 2: Map the Data Elements

The next step in transforming the pallet elevator DES into a DT involves mapping the data elements. It is crucial to identify the data sources required to integrating real-time data feeds, sensors, and other sources of information to construct a virtual counterpart of the system [253]. To identify the necessary data sources, the first step is to map the data used as input to represent the system in the DES model. This involves understanding the existing data inputs and their significance in capturing the system’s behavior and characteristics. Then, the data elements that need to be modified or additionally captured to represent and connect with real-time data were identified.

A lack of real-time data makes a simulation static, while a DT is active [91]. A simulation model is considered static until it starts receiving real-time data from its real-world counterpart [14]. Once that happens, it becomes active and is considered a DT. Therefore, it is important to identify the components and data elements that affect the process, and to categorize which elements are either static or dynamic. Static components are those that do not change over time or are unaffected by real-time data [91]. Dynamic components on the other hand, are influenced by real-time data and exhibit varying behavior based on the information received [91]. By categorizing the components as static or dynamic, we can determine which aspects of the DT require real-time data integration and which elements can remain static.

The existing pallet elevator DES system consists of several components, including two assembly machines, one elevator, pallets, and conveyors responsible for transporting the pallets. Table 51 provides a comprehensive overview of the data types utilized within this system. These data types include processing times for the machines, conveyor capacities, and other relevant parameters. Static data refers to information that remains the same throughout the transformation from the DES to a DT. For example, the capacity of the elevator, which is always set to 1 pallet and does not change, is classified as static data. These static data elements retain their fixed values within the DT. On the other hand, dynamic data elements need to be modified to accommodate real-time data feeds in the DT.

For example, in the DES, the interarrival time was represented by a data table that was used as historical data input. However, in the DT, this dynamic data element will be updated to incorporate real-time data collected from sensors. To facilitate the integration of the data table with the simulation model, an automatic binding process will be implemented. This process involves linking the data table to the simulation model through an automatic binding mechanism using a

.csv file. This binding process ensures that any updates or changes made to the data table will be automatically reflected in the simulation model. It enables real-time synchronization between the data table and the simulation, allowing for accurate and up-to-date data inputs. Certain data and components related to the system were changed and represented differently. For instance, elements such as reliability logic, which includes downtimes from failures and maintenance, were incorporated into the DT model as a work schedule for the machines.

Table 51 Data Element Mapping in DES and DT.

Element	**Data Type**	**Static/Dynamic**	**DES**	**DT**
Source 1	Interarrival Time	Dynamic	Arrival Table	.csv table
Source 2	Deliver pallets at the beginning of the run	Dynamic	9 Pallets	.csv table
Source 3	Deliver pallets at the beginning of the run	Dynamic	9 Pallets	.csv table
Head Load B	Processing Time	Static	21.7 seconds	21.7 seconds
	Work Schedule	Dynamic	Shifts	.csv table
	Reliability Logic	Dynamic	Random Distribution	Represented by the timestamps of the work schedule
Head Install	Processing Time	Static	28.1 seconds	28.1 seconds
	Work Schedule	Dynamic	Shifts and Breaks	.csv table
	Reliability Logic	Dynamic	Random Distribution	Represented by the timestamps of the work schedule
Elevator	Capacity	Static	1 pallet	1 pallet
	Length	Static	4 meters	4 meters
	Speed	Static	45 meters/second	45 meters/second
	Load/Unload times	Static	4 second	4 second
	Logic	Static	Current logic	Current logic
Conveyors	Capacity	Static	Constant capacity	Constant capacity
	Length	Static	Constant length	Constant length
	Speed	Static	21 meters/ minute	21 meters/minute
Pallets (Entity)	Routing Logic	Static	Sequence table	Sequence table
	Speed	Static	21 meters/ minute	21 meters/minute

For the dynamic data, the simulation model will periodically receive a list of data specifically identified as necessary for reflecting the real process. This data, which includes the interarrival time, the number of pallets present in the buffer at the current state of the system, and both machines Head Load B and Head Install work schedules timestamp, would be updated regularly to ensure the simulation model remains synchronized with the current state of the system. By focusing on the dynamic elements that significantly influence the system’s behavior, this approach allows for a near real-time DT that captures relevant changes and updates in a timely manner.

6.4.1.3 Step 3: Create a System Architecture

As the DT receives real-time performance data from the physical system, synchronization between the DT and its physical counterpart is necessary to ensure that the production systems are continuously updated [250]. Instead of performing input analysis for the simulation model, which involves describing the input random variables and providing the simulation program with the appropriate distributions and processes [39], a real-time system architecture was developed for data collection, storage, and connection. System architecture refers to the formal description and representation of a system, emphasizing its structure and behaviors [254,255]. In the context of this research, the system architecture comprises four main components: the physical environment, the digital environment, the cloud service, and the digital interface. Figure 39 illustrates the system architecture employed to collect, store, and process the information obtained from the sensors.

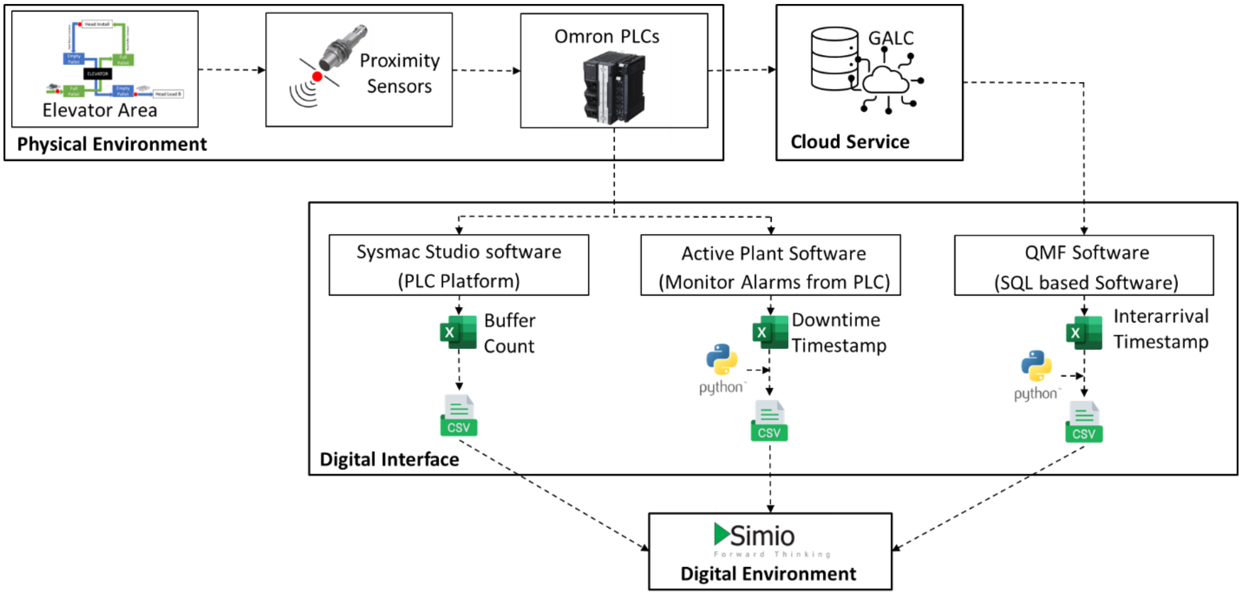


Figure 39 System Architecture of the DT.

The physical environment component represents the tangible aspects of the system, including the physical machines, conveyors, pallets, sensors, PLCs, and other physical elements involved in the pallet elevator system. The physical environment encompasses the real-world components and processes that the DT aims to replicate [256].

The digital environment refers to the virtual counterpart of the physical system [3]. It includes simulation software chosen for this research, Simio, that provides a digital representation of the pallet elevator system. The digital environment defines how humans can engage and interact with the digital system [10].

The cloud service component involves the utilization of cloud-based storage and computing resources [257]. It serves as a scalable and flexible platform for storing and processing substantial amounts of data generated by the system. The cloud service provides the necessary infrastructure to handle data storage, retrieval, and analysis, ensuring efficient data management for the DT [90]. In this research, the cloud component used by the automotive manufacturing company is referred to as GALC, which is an internal cloud system specific to the company designed to meet their unique requirements and cater to their automotive manufacturing processes.

The digital interface serves as the means of communication and interaction between the physical and digital components of the system [139]. It enables the collection of data from sensors, facilitates the transfer of data to simulation software, and allows users to interact with the DT. The purpose of the interface is to ensure that as the real process changes, the model is updated periodically [97]. In this research, the digital interface includes software applications, such as Query Management Facility (QMF) and Active Plant software, and programming interfaces, such as Python scripts, which enable seamless integration and communication between the physical and digital environments.

As identified earlier, the data that requires regular updates in the DT includes the interarrival time, the number of pallets in the buffer, and the timestamps for the work schedules of both machines, Head Load B and Head Install, which include downtimes caused by failures, starving, or blocking. By including these downtimes in the work schedules, a comprehensive record of the machines' activities and non-productive periods is ensured. Proximity sensors are strategically positioned at various locations within the designated area to capture the necessary data. These sensors automatically detect and transmit information to PLCs. PLCs are computerized devices widely used in industrial settings for controlling machinery and automating entire systems [258,259]. The term “logic” is employed because the programming primarily focuses on executing logical and switching operations [259]. PLCs function as central control units that receive input signals from control devices such as proximity sensors and execute instructions defined in the user’s ladder program [260]. The ladder program provides instructions to the PLCs, guiding it on how to respond to the received signals [259]. In this context, the PLCs serve as the interface between the machines, conveyors, pallet elevator, and sensors. They facilitate the interaction and communication among these components, ensuring that data flows smoothly within the system. PLCs have been in use since the late 1960s, first used by the automotive industry. Over time, they have become a common fixture in various industries [261], including the automotive manufacturing company under study in this research. Given their widespread usage and compatibility with the machinery and sensors involved in the pallet elevator system, PLCs were selected as the ideal technology for integrating and interacting with these components.

Three types of data are obtained from the PLCs and each type is accessed through three specific software applications:

- Number of pallets in the buffer data: This data is retrieved from the PLCs using the Sysmac Studio software. Sysmac Studio serves as the platform for programming and configuring the PLCs, allowing access to relevant data points such as the number of pallets present in the buffer. Through Sysmac Studio, the DT can collect and monitor this data in real-time, enabling an accurate representation of the system's state.
- Work schedule data: The work schedule data, which incorporates the timestamp of operations and downtime, pertaining to both machines, Head Load B and Head Install, is accessed from the PLCs using the Active Plant software. Active Plant software serves as a dedicated tool for managing and visualizing production-related data. It provides a user-friendly interface to retrieve and analyze work schedule information, allowing the DT to align its virtual work schedules with the actual operations of the physical machines.
- Interarrival time data: The interarrival time data is sent from the PLCs to the cloud service (GALC) for storage. To access this stored data, the QMF software is utilized. QMF serves as a data retrieval and query tool, specifically designed to interact with cloud-based

databases. It allows the DT to retrieve the interarrival time data from the cloud storage, enabling analysis and integration of real-time interarrival times into the simulation model.

After retrieving the data from the PLCs using the Sysmac Studio, Active Plant, and QMF software, the collected data is stored in Excel format. However, to properly integrate and utilize this data within the simulation software (Simio), data manipulation processes are required. To accomplish this, Python codes were developed to automatically format the collected data into the appropriate format, specifically a comma-separated values (.csv) format. This formatting ensures compatibility with the Simio simulation software. By creating custom Python scripts, the collected data is processed and transformed into the appropriate format compatible with Simio. This ensures seamless integration of the real-time data into the simulation model, enabling accurate representation and analysis of the system dynamics within Simio's simulation environment. This connection is set up as a data table input that is bound to an external .csv file, enabling the virtual model to be automatically updated and simulated with the operational data. After binding the table to a particular file, in this case a .csv file, the importing options can be selected, which can be manual (by clicking the Import button at the beginning of each simulation run) or automatic during each model run. The automatic import is useful when the data undergoes frequent changes, such as initializing the model to reflect the current system status, which is the case for this research [39].

6.4.1.4 Step 4: Collect the Data

Data collection is a critical aspect of the Representation phase as it forms the foundation for constructing the digital model [262]. Subsequent phases in the 4R framework will build upon this data foundation. In this research, the data collection process for the three types of data mentioned earlier follows the established system architecture. As mentioned before, the pallet elevator area is already automated, which means that the data collection occurs seamlessly as part of the existing automation infrastructure. The industry partners were consulted regarding this, and they explained that the data had been previously validated many times. Since the pallet elevator area is already automated, the data collection process is integrated into the functioning of the system. The data is obtained directly from the PLCs and cloud service through the designated software applications.

During the Representation phase, the primary objective is to convert the essential variables and characteristics of the physical system into a digital format. This process entails leveraging various tools and technologies, such as sensors, applications, and digital interface tools, to gather and store the relevant data. The digital interface served as the bridge between the simulation model and the real process, enabling the integration of both and achieving the first capability level of the 4R framework, Representation (R1). By following these steps, this research successfully documented the process of transforming the DES into the Representation level of a DT and laid the groundwork for further exploration and advancements in the subsequent levels of the 4R framework.

6.4.2 Achieving Replication (R2)

To achieve the Replication (R2) level of capability in a DT based on a DES, the process initiated with the mapping of elements within the virtual environment to assess whether the existing elements derived from the DES were adequate to represent the intended DT. Subsequently, the integration of real-time data sources from the architecture established during the Representation phase into the DT took place. Finally, a comprehensive verification and validation process was conducted to ensure that the DT could generate equivalent outputs when presented with the same inputs as the physical system. A detail description of each of these steps are provided below.

6.4.2.1 Map the Virtual Environment Elements

In this step, the evaluation of whether the existing elements in the virtual system derived from the existing DES are sufficient to represent the desired DT was performed. The mapping exercise involved a careful examination of the elements in the existing DES and their correspondence to the key variables and characteristics of the physical system. This included factors such as processing times, capacities, interarrival times, work schedules, and any other relevant aspects.

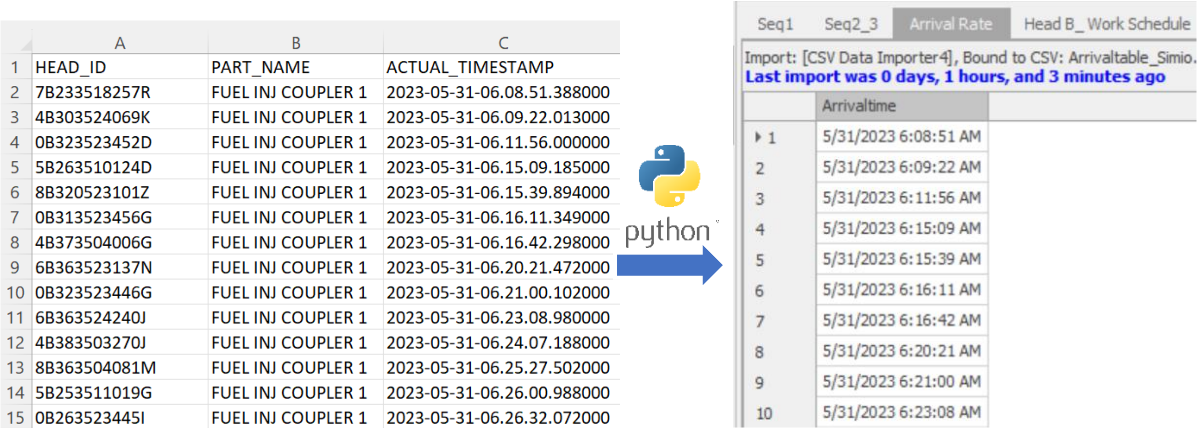
In this case study, the only modification implemented to the Simio model was the removal of an additional server previously utilized to establish a work schedule for the system's source. This server became unnecessary in the DT environment since the arrival table, which automatically connected to the physical system’s data, already incorporates the off-shift times. After making the necessary modifications, the virtual system is now encompassing the necessary functionality to replicate and represent the behavior and performance of the physical system.

6.4.2.2 Integrate the Simulation Model and Data Sources

In the Representation level of the DT transformation process, it was identified that certain data types within the existing DES would need to be updated to incorporate real-time data and enable dynamic behavior.

One of the identified data types is the interarrival time, which was previously represented by a data table based on historical data in the DES. The interarrival time will now be represented as a data table that contains real-time data collected from sensors and is automatically updated. A Simio data table resembles a spreadsheet table in its structure and offers features such as import, export, and binding to external files [39]. Access to data within these tables can be done sequentially, randomly, directly, or automatically. In this case study, the data table that contains the interarrival time of pallets in the system is bound to the simulation model as a .csv file and the data is imported automatically each time a model run is started, allowing the DT to continuously receive and incorporate updated interarrival times, reflecting the actual arrival patterns of pallets in the system, up to the time the model is run.

The interarrival time data is transmitted from the PLCs on the line to the cloud for storage. The QMF software is employed to access this stored data. QMF serves as a tool for data retrieval and querying, specifically designed to interact with databases hosted in the cloud. Subsequently, the data is converted from an Excel file to the appropriate format (.csv) using Python, which automatically processes the original data and converts it into the appropriate format for binding to Simio. Figure 40 illustrates the initial Excel file containing the interarrival timestamps gathered from sensors, alongside the transformed table, in Simio, which is bound to a .csv file. The python script responsible for executing this transformation is depicted as a pseudocode in Table 52.



	A	B	C
1	HEAD_ID	PART_NAME	ACTUAL_TIMESTAMP
2	7B233518257R	FUEL INJ COUPLER 1	2023-05-31-06.08.51.388000
3	4B303524069K	FUEL INJ COUPLER 1	2023-05-31-06.09.22.013000
4	0B323523452D	FUEL INJ COUPLER 1	2023-05-31-06.11.56.000000
5	5B263510124D	FUEL INJ COUPLER 1	2023-05-31-06.15.09.185000
6	8B320523101Z	FUEL INJ COUPLER 1	2023-05-31-06.15.39.894000
7	0B313523456G	FUEL INJ COUPLER 1	2023-05-31-06.16.11.349000
8	4B373504006G	FUEL INJ COUPLER 1	2023-05-31-06.16.42.298000
9	6B363523137N	FUEL INJ COUPLER 1	2023-05-31-06.20.21.472000
10	0B323523446G	FUEL INJ COUPLER 1	2023-05-31-06.21.00.102000
11	6B363524240J	FUEL INJ COUPLER 1	2023-05-31-06.23.08.980000
12	4B383503270J	FUEL INJ COUPLER 1	2023-05-31-06.24.07.188000
13	8B363504081M	FUEL INJ COUPLER 1	2023-05-31-06.25.27.502000
14	5B253511019G	FUEL INJ COUPLER 1	2023-05-31-06.26.00.988000
15	0B263523445I	FUEL INJ COUPLER 1	2023-05-31-06.26.32.072000

Import: [CSV Data Importer4], Bound to CSV: Arrivaltable_Simio.
Last import was 0 days, 1 hours, and 3 minutes ago

	Arrivaltime
1	5/31/2023 6:08:51 AM
2	5/31/2023 6:09:22 AM
3	5/31/2023 6:11:56 AM
4	5/31/2023 6:15:09 AM
5	5/31/2023 6:15:39 AM
6	5/31/2023 6:16:11 AM
7	5/31/2023 6:16:42 AM
8	5/31/2023 6:20:21 AM
9	5/31/2023 6:21:00 AM
10	5/31/2023 6:23:08 AM

Figure 40 Interarrival Times Transformed by Python and Bound to Simio.

Table 52 Python Script (Pseudocode) to Transform the Interarrival Data from Sensors.

Pseudocode of the Python Script to Transform the Interarrival Data from Sensors

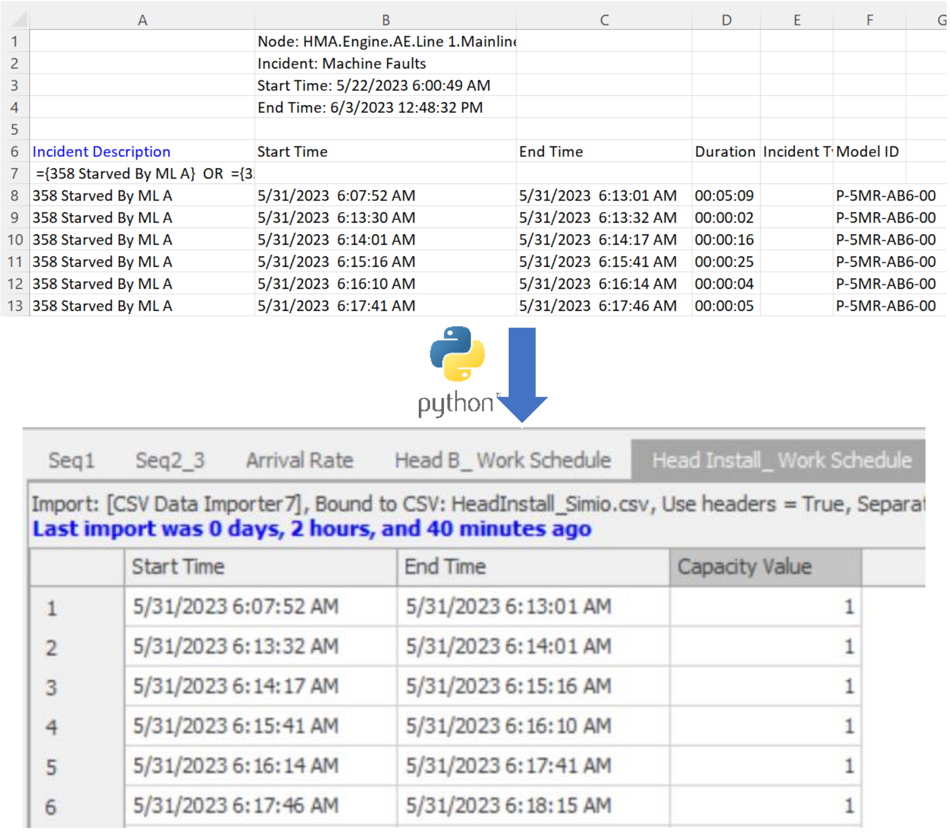
```
Import pandas library
Import datetime module from datetime library
Define function transform_datetime(value):
    Replace "." with "-" in the value
    Split the value by "-" into a list called date_list
    Extract year, month, day, hour, minute, second, and microsecond from date_list
    Create a datetime object using the extracted components
    Store the formatted datetime as formatted_datetime
    Return formatted_datetime
Read the data from the Excel file using pandas and store it in a dataframe called df
Create an empty dataframe called new_df
Apply the transform_datetime function to the 'ACTUAL_TIMESTAMP' column of df
Store the transformed values in a new column named 'Arrivaltime' in new_df
Export new_df dataframe to a CSV file named name_newfile
```

Another data type that requires regular updating is the number of pallets present in the buffer at the current state of the system. In the DES, this data was static and did not change over time. However, in the DT, it is crucial to have dynamic data that accurately represents the real-time status of the buffer. This can be achieved by connecting the buffer to the data feed from sensors or other monitoring devices, ensuring that the DT receives updated information on the number of pallets in the buffer. In the pallet elevator area, there are two pallet buffers: Head return and Head buffer. The number of pallets present in the buffers is captured from sensors and utilized as input. This data is connected to data tables that update the buffer quantities in real-time at the beginning of each run, ensuring an accurate simulation of the system’s behavior and dynamics.

Additionally, the work schedules of both machines, Head Load B and Head Install, were identified as important data types to be updated. The timestamps of these work schedules should reflect real-time data to capture the current status and progress of the operations. The timestamps associated with the Head Install machine correspond to the periods of downtime that occur as a result of issues upstream. These issues lead to blockages or delays that are not directly caused by the Head Install machine itself. Similar to the Head Install machine, the downtime of the Head Load B machine is also caused by problems downstream. These issues occurring further along the production line result in delays or disruptions that affect the Head Load B machine's operation and contribute to its downtime.

Figure 41 displays the original Excel file that includes the recorded downtimes obtained from sensors, specifically for the Head Install machine. Additionally, it displays the corresponding transformed table, which is bound to a .csv file in Simio. This downtime data is accessed from PLCs using the Active Plant software. Like the interarrival times, a python script is employed to automatically process the original data and convert it into the appropriate format for integration with Simio. In this case, the data collected from sensors pertains to the downtime of the Head Install machine. However, to utilize this data as a work schedule for the machine within the Simio model, it needs to be converted into machine uptime. The python script depicted as a pseudocode in Table 53 is responsible for executing this transformation. An identical process, utilizing the same python script, is applied to downtime data for the Head Load B machine. The Python script performs the necessary adjustments to convert the recorded downtime into machine uptime, which determines the periods when the machine is operational and available for work in the Simio simulation. By employing the python script, the process of converting downtime to machine uptime is automated, eliminating the need for manual data manipulation.

The python scripts developed specifically for this DES to DT transformation play a crucial role in facilitating the integration of the sensor data into the Simio simulation model. They serve as a bridge between the raw data obtained from the sensors and the required format for Simio’s data tables. By automating this transformation process, the python scripts simplify the data preprocessing step, saving time and effort that would otherwise be spent on manual data manipulation.



	A	B	C	D	E	F	G
1		Node: HMA.Engine.AE.Line 1.Mainlin					
2		Incident: Machine Faults					
3		Start Time: 5/22/2023 6:00:49 AM					
4		End Time: 6/3/2023 12:48:32 PM					
5							
6	Incident Description	Start Time	End Time	Duration	Incident T	Model ID	
7	={358 Starved By ML A} OR ={3						
8	358 Starved By ML A	5/31/2023 6:07:52 AM	5/31/2023 6:13:01 AM	00:05:09		P-5MR-AB6-00	
9	358 Starved By ML A	5/31/2023 6:13:30 AM	5/31/2023 6:13:32 AM	00:00:02		P-5MR-AB6-00	
10	358 Starved By ML A	5/31/2023 6:14:01 AM	5/31/2023 6:14:17 AM	00:00:16		P-5MR-AB6-00	
11	358 Starved By ML A	5/31/2023 6:15:16 AM	5/31/2023 6:15:41 AM	00:00:25		P-5MR-AB6-00	
12	358 Starved By ML A	5/31/2023 6:16:10 AM	5/31/2023 6:16:14 AM	00:00:04		P-5MR-AB6-00	
13	358 Starved By ML A	5/31/2023 6:17:41 AM	5/31/2023 6:17:46 AM	00:00:05		P-5MR-AB6-00	

Import: [CSV Data Importer7], Bound to CSV: HeadInstall_Simio.csv, Use headers = True, Separa

Last import was 0 days, 2 hours, and 40 minutes ago

	Start Time	End Time	Capacity Value
1	5/31/2023 6:07:52 AM	5/31/2023 6:13:01 AM	1
2	5/31/2023 6:13:32 AM	5/31/2023 6:14:01 AM	1
3	5/31/2023 6:14:17 AM	5/31/2023 6:15:16 AM	1
4	5/31/2023 6:15:41 AM	5/31/2023 6:16:10 AM	1
5	5/31/2023 6:16:14 AM	5/31/2023 6:17:41 AM	1
6	5/31/2023 6:17:46 AM	5/31/2023 6:18:15 AM	1

Figure 41 Head Install Downtimes Transformed by Python and Bound to Simio.

Table 53 Python Script (Pseudocode) to Transform the Head Install data from Sensors.

Pseudocode of the Python Script to Transform the Head Install data from Sensors

```
Import pandas library
Import datetime module from datetime library
Read the data from the Excel file using pandas and store it in a dataframe called df
Remove the first 7 rows from the dataframe by using iloc[6:] to capture data from the previous row
Delete the last two columns from the dataframe by using iloc[: :-2]
Rename the columns of the dataframe to "Incident Description", "Start Time", "End Time", and "Duration"
Filter the dataframe based on the condition where "Incident Description" is equal to "360 Blocked By ML A" and "358 Starved By ML A"
Reset the index of the dataframe by using reset_index(drop=True)
Create a new dataframe called df_new
Assign the "End Time" values from the previous row (excluding the last row) to the "StartTime" column of df_new
Assign the "Start Time" values from the next row (excluding the first row) to the "EndTime" column of df_new, resetting the index
Add a new column named "CapacityValue" to df_new with a value of 1 for all rows
Export the df_new dataframe to a CSV file named new_file_name, excluding the index
```

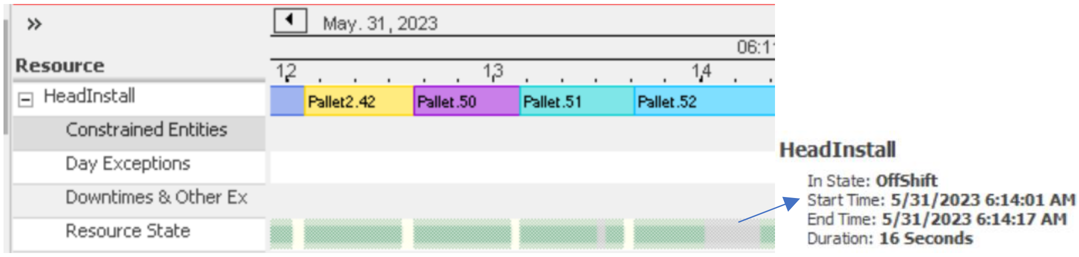
6.4.2.3 Verification and Validation of the DT

To confirm that the model has achieved the R2 level of DT, verification and validation are performed. The primary focus of this step is to ensure that the DT is generating equivalent outputs when presented with the same inputs as the physical system.

In accordance with section 6.4.2.2, real-time data is gathered from sensors and utilized as data tables. To verify and validate the model, a single day's worth of timestamps (specifically, 05/31/2023) was captured by the sensors and automatically connected to the model. The objective of this step is to execute the DT model for that day and compare the simulated events with the actual events that occurred during that time period. To accomplish this, completion time-stamps from the two machines in the system were collected and utilized to validate the outcomes generated by the model. On that particular day, the operational period spanned from 6:06:48 am to 11:59:59 pm, encompassing two shifts.

For verification and validation purposes, the Simio Planning Tab was employed to generate a time-stamp timeline. This timeline was created using a Plan that comprised both a Resource Plan and an Entity Workflow. The Resource Plan provided detailed information about the servers, including the pallets being processed and server state (off shift, processing, starving, etc.). Work schedules were verified by observing the times that the machines are in off-shift periods and comparing it to the work schedule table. It is important to remember that work schedules are employed to reflect the reliability logic of the machines. During downtime, when a machine is not operational, it is considered off-shift, whereas when it is actively processing, it is considered on-shift.

Figure 42 illustrates an example of the Resource Plan, specifically for the Head Install server. Adjacent to the Resource State row, several colors represent the state of this machine: green denotes processing, yellow denotes blocking, while grey denotes off-shift. To verify if the off-shift periods, in grey, align with the designated timings in the work schedule, the start and end times were compared. For example, according to the work schedule, Head install is off-shift from 6:14:01am to 6:14:17am, which is exactly what shows in the Resource Plan. Therefore, the work schedule for Head Install was verified.

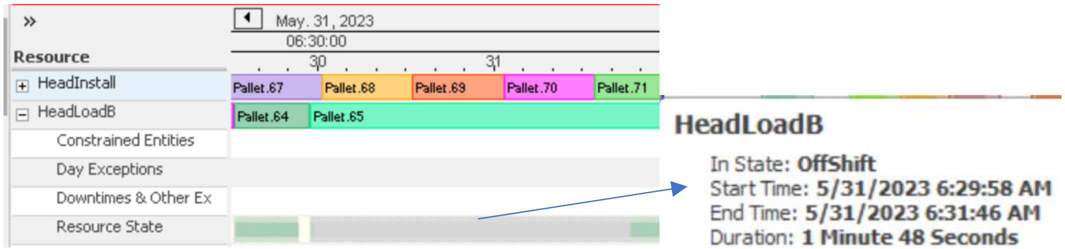


HeadInstall

In State: **OffShift**
Start Time: **5/31/2023 6:14:01 AM**
End Time: **5/31/2023 6:14:17 AM**
Duration: **16 Seconds**

Figure 42 Example of Resource Plan for Head Install.

The same process was performed to ensure that the other machine in the system, Head Load B, was performing as is expected according to the work schedule tables. Figure 43 illustrates the start and end times of one instance of off shift. This time was compared and verified against the work schedule table.

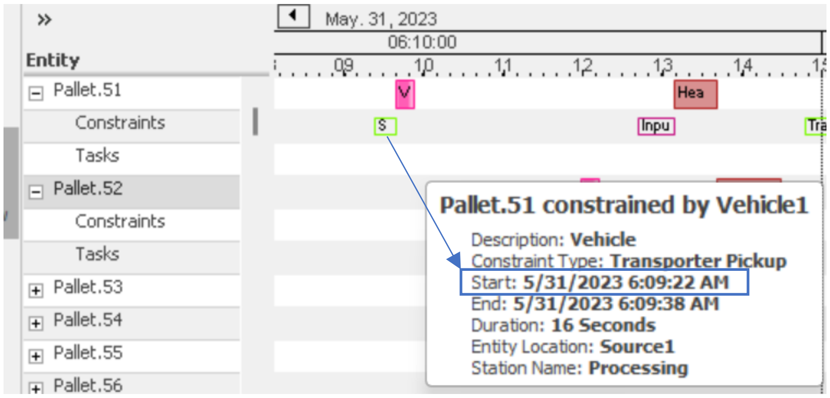


HeadLoadB

In State: **OffShift**
Start Time: **5/31/2023 6:29:58 AM**
End Time: **5/31/2023 6:31:46 AM**
Duration: **1 Minute 48 Seconds**

Figure 43 Example of Resource Plan for Head Load B.

The Entity Workflow was utilized to verify the arrival times, processing times, and completion times of each pallet by comparing the observed values in the Entity Workflow with the expected values in the Arrival Table. For instance, Figure 44 illustrates that a pallet arrived in the system at 6:09:22 am, aligning with the expected arrival time stated in the Arrivals Table. Consequently, the order's arrival time was verified to be as intended.

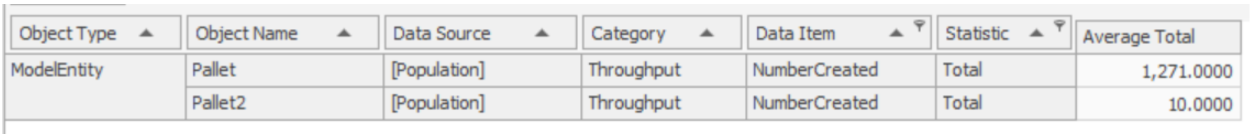


Pallet.51 constrained by Vehicle1

Description: **Vehicle**
Constraint Type: **Transporter Pickup**
Start: **5/31/2023 6:09:22 AM**
End: **5/31/2023 6:09:38 AM**
Duration: **16 Seconds**
Entity Location: **Source1**
Station Name: **Processing**

Figure 44 Example of Entity Workflow.

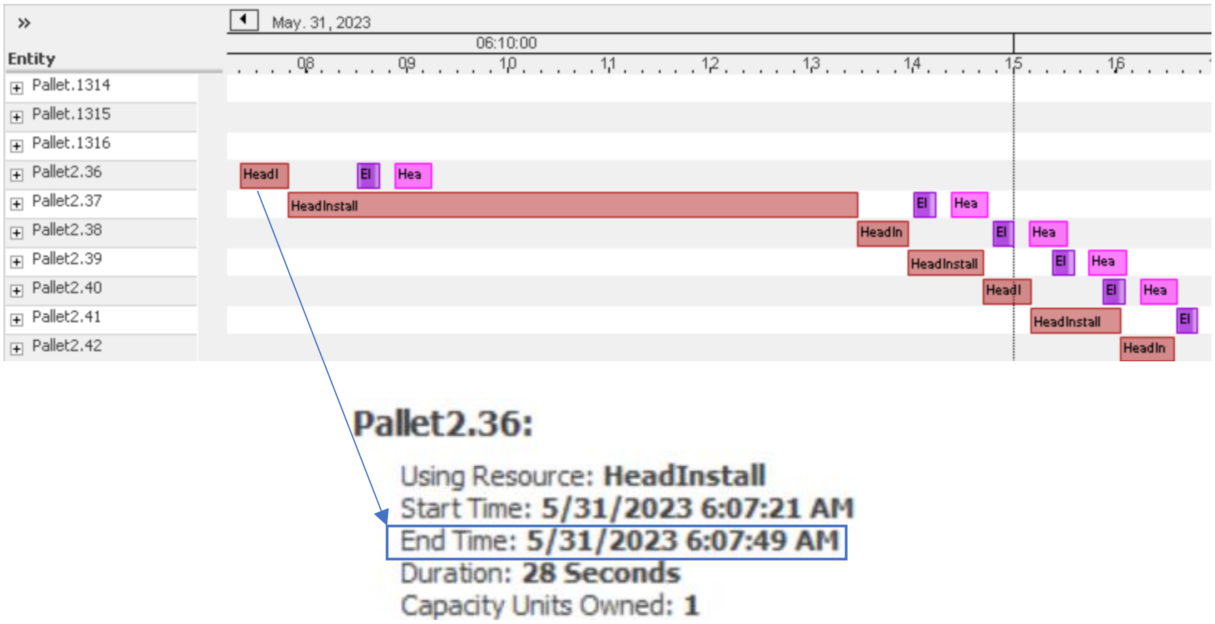
As previously mentioned, the system in question consists of three sources of pallets. Based on the arrival table, the total number of pallet instances that arrived on this specific day was recorded as 1271 pallets. This number aligns precisely with the calculated number of entities (pallets) generated within the simulation system, as depicted in Figure 45.



Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average Total
ModelEntity	Pallet	[Population]	Throughput	NumberCreated	Total	1,271.0000
	Pallet2	[Population]	Throughput	NumberCreated	Total	10.0000

Figure 45 Number Created in the System.

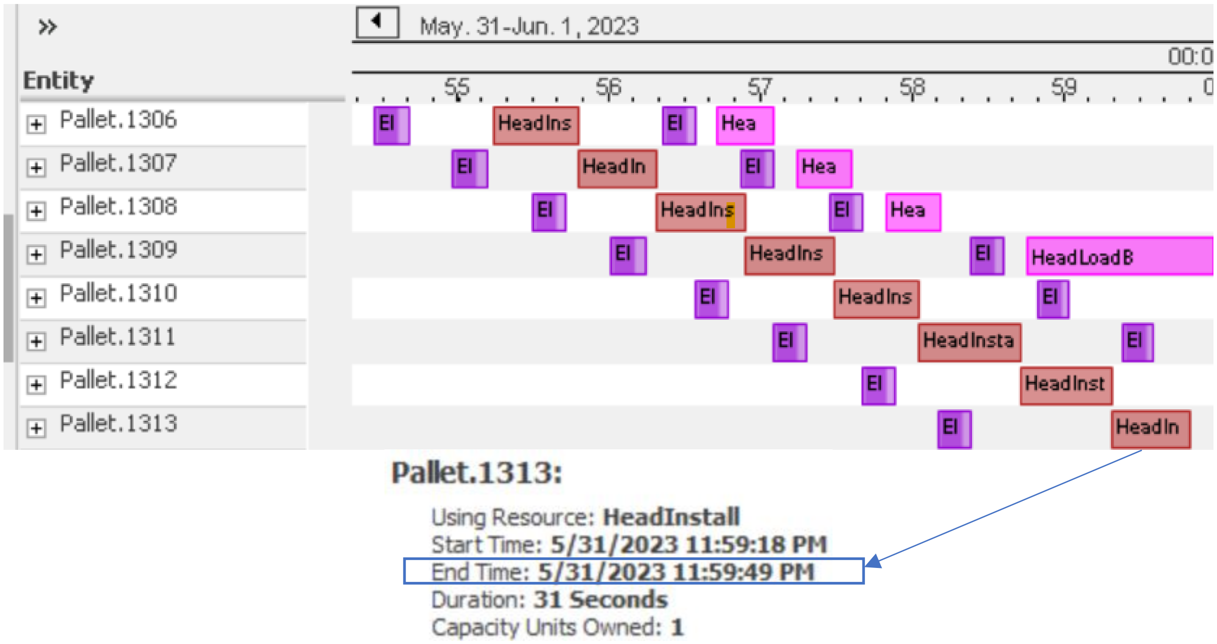
To validate the model, the completion timestamps gathered from the physical system's Head Install machine are compared with the timestamps derived from the virtual model. Figure 46 exhibits the completion timestamps from the Head Install machine, with the initial event completed at 6:07:49 am highlighted in green, and the final event completed at 11:59:49 pm highlighted in red. These timestamps are then compared with the Resource Plan acquired from the simulation, which exhibits a perfect match. Figure 46 depicts the completion timestamp of the first pallet within the system, recorded at 6:07:49 am, while Figure 47 showcases the completion timestamp of the last pallet in the system, occurring at 11:59:40 pm.



Pallet2.36:

Using Resource: **HeadInstall**
Start Time: **5/31/2023 6:07:21 AM**
End Time: **5/31/2023 6:07:49 AM**
Duration: **28 Seconds**
Capacity Units Owned: **1**

Figure 46 Completion Timestamp of the First Pallet in the System.



Pallet.1313:

Using Resource: **HeadInstall**
Start Time: **5/31/2023 11:59:18 PM**
End Time: **5/31/2023 11:59:49 PM**
Duration: **31 Seconds**
Capacity Units Owned: **1**

Figure 47 Completion Time of the Last Pallet in the System.

It is worth noting that while the completion timestamps of the first and last pallets aligned perfectly with the expected times, there were slight deviations in the timestamps of the pallets in between. These deviations amounted to a difference of an average of 3.5 seconds from the expected completion times, as is shown in Table 54. These deviations occur due to the presence of static data in the model, as indicated in Table 51. Specifically, the processing times for the machines in the model remained static when transforming from a DES to a DT because the sensors currently do not record processing timestamps. As a result, considering the input and insights obtained from the managers of this area, it was concluded that the slight variations observed in the completion times, attributed to the presence of static values, remained indicative of the typical performance. Therefore, these deviations were deemed acceptable, and the model was considered valid. With the sensors currently installed, it is impossible to get exact processing times, therefore, getting an exact match in these times is not possible, at this time.

Table 54 Completion Timestamp Comparison for Head Install.

Head Install			
PALLET ID	**REAL COMPLETION TIMESTAMP**	**SIMULATED COMPLETION TIMESTAMP**	**ABSOLUTE ERRORS (SECONDS)**
J35Y76073507	2023-05-31 6:07:49 am	2023-05-31 6:07:49 am	0
J35Y76073503	2023-05-31 6:13:27 am	2023-05-31 6:13:26 am	1
J35Y76073508	2023-05-31 6:13:58 am	2023-05-31 6:13:56 am	2
J35Y76073509	2023-05-31 6:14:43 am	2023-05-31 6:14:40 am	3
J35Y76073510	2023-05-31 6:15:13 am	2023-05-31 6:15:09 am	4
J35Y76073511	2023-05-31 6:16:07 am	2023-05-31 6:16:02 am	5
J35Y76073512	2023-05-31 6:16:40 am	2023-05-31 6:16:34 am	6
J35Y76073505	2023-05-31 6:17:09 am	2023-05-31 6:17:02 am	7
⋮	⋮	⋮	⋮
J35Y81054549	2023-05-31 2:13:10 pm	2023-05-31 2:13:04 pm	6
J35Y81054550	2023-05-31 2:13:43 pm	2023-05-31 2:13:38 pm	5
J35Y81054551	2023-05-31 2:14:17 pm	2023-05-31 2:14:13 pm	4
J35Y81054548	2023-05-31 2:15:43 pm	2023-05-31 2:15:38 pm	5
J35Y81054552	2023-05-31 2:16:18 pm	2023-05-31 2:16:11 pm	7
J35Y81054555	2023-05-31 2:16:48 pm	2023-05-31 2:16:40 pm	8

J35Y81054546	2023-05-31 2:17:17 pm	2023-05-31 2:17:09 pm	8
J35Y81054556	2023-05-31 2:17:49 pm	2023-05-31 2:17:41 pm	8
⋮	⋮	⋮	⋮
J35Y68744060	2023-05-31 11:55:48 pm	2023-05-31 11:55:47 pm	1
J35Y68744043	2023-05-31 11:56:19 pm	2023-05-31 11:56:18 pm	1
J35Y68744062	2023-05-31 11:56:54 pm	2023-05-31 11:56:53 pm	1
J35Y68744041	2023-05-31 11:57:29 pm	2023-05-31 11:57:28 pm	1
J35Y68744065	2023-05-31 11:58:02 pm	2023-05-31 11:58:01 pm	1
J35Y68744045	2023-05-31 11:58:42 pm	2023-05-31 11:58:42 pm	0
J35Y68744067	2023-05-31 11:59:18 pm	2023-05-31 11:59:18 pm	0
J35Y68744050	2023-05-31 11:59:49 pm	2023-05-31 11:59:49 pm	0

A similar process was conducted for the Head Load B machine, and in this case, the differences in completion timestamps between the physical and virtual environments were slightly larger (average of 11.7 seconds) compared to the Head Install machine, as shown in Table 55. However, according to the managers of this area, these differences were deemed sufficiently close for validation, as the primary focus was on the Head Install machine, where the times matched much more closely.

Table 55 Completion Timestamp Comparison for Head Load B.

Head Load B			
PALLET ID	**REAL COMPLETION TIMESTAMP**	**SIMULATED TIMESTAMP**	**ABSOLUTE ERRORS (SECONDS)**
J35Y76073539	2023-05-31 6:09:50 am	2023-05-31 6:09:14 am	36
J35Y76073540	2023-05-31 6:15:10 am	2023-05-31 6:14:43 am	29
J35Y76073541	2023-05-31 6:15:42 am	2023-05-31 6:15:30 am	12
J35Y76073542	2023-05-31 6.16.40 am	2023-05-31 6:16:35 am	5
J35Y76073543	2023-05-31 6.17.10 am	2023-05-31 6:17:19 am	9
J35Y76073544	2023-05-31 6.17.51 am	2023-05-31 6:17:51 am	0
J35Y76073545	2023-05-31 6.18.24 am	2023-05-31 6:18:27 am	3

J35Y76073546	2023-05-31 6.19.02 am	2023-05-31 6:18:55 am	7
⋮	⋮	⋮	⋮
J35Y81054572	2023-05-31 2:09:28 pm	2023-05-31 2:09:24 pm	4
J35Y81054573	2023-05-31 2:10:01 pm	2023-05-31 2:09:57 pm	4
J35Y81054574	2023-05-31 2:10:33 pm	2023-05-31 2:10:30 pm	3
J35Y81054575	2023-05-31 2:11:04 pm	2023-05-31 2:11:03 pm	1
J35Y81054576	2023-05-31 2:11:39 pm	2023-05-31 2:11:37 pm	2
J35Y81054577	2023-05-31 2:12:09 pm	2023-05-31 2:12:18 pm	9
J35Y81054578	2023-05-31 2:12:41 pm	2023-05-31 2:12:44 pm	3
J35Y81054579	2023-05-31 2:13:14 pm	2023-05-31 2:13:16 pm	2
⋮	⋮	⋮	⋮
J35Y68744081	2023-05-31 11:56:26 pm	2023-05-31 11:56:00 pm	26
J35Y68744082	2023-05-31 11:56:54 pm	2023-05-31 11:56:31 pm	23
J35Y68744083	2023-05-31 11:57:24 pm	2023-05-31 11:57:04 pm	20
J35Y68744084	2023-05-31 11:57:54 pm	2023-05-31 11:57:35 pm	19
J35Y68744085	2023-05-31 11:58:24 pm	2023-05-31 11:58:10 pm	14
J35Y68744086	2023-05-31 11:58:54 pm	2023-05-31 11:58:35 pm	19
J35Y68744087	2023-05-31 11:59:25 pm	2023-05-31 11:58:55 pm	20
J35Y68744088	2023-05-31 11:59:56 pm	2023-05-31 11:59:59 pm	3

Table 56 provides a comparison of starved timestamps for the Head Install machine, showcasing the comparison between real data from the physical system and simulated data from the model. The simulated data start times (highlighted in blue) occur a few seconds ahead of the expected data (highlighted in grey). The reason for this variation, once again, stems from the presence of static data elements in the model, namely the processing times for machines, elevator operations, and conveyor speed. Despite the slight time difference, the managers of the area have validated this result. They have agreed that the duration of the starved times remains nearly identical, as evidenced by the data in Table 56. It is worth noting that there were additional starting events with durations as short as between 1 and 10 seconds. Despite the small differences in their durations, the overall trend and consistency of the starting events align between the simulated data and the real data from the physical system. This reinforces the validation of the model's performance and its ability to capture the general behavior of the system accurately.

Table 56 Head Install Starved Timestamp Comparison.

Real Data Start Time	Duration (Sec)	Simulated Data Start Time	Duration (Sec)	Relative Error (%)
5/31/2023 6:23:40 am	49	5/31/2023 6:23:11 am	40	18%
5/31/2023 6:24:57 am	30	5/31/2023 6:24:19 am	30	0%
5/31/2023 6:25:56 am	54	5/31/2023 6:25:18 am	51	6%
5/31/2023 6:35:49 am	25	5/31/2023 6:35:20 am	15	40%
5/31/2023 6:46:34 am	70	5/31/2023 6:45:57 am	69	1%
5/31/2023 6:49:14 am	13	5/31/2023 6:48:44 am	13	0%
⋮	⋮	⋮	⋮	⋮
5/31/2023 7:38:06 am	106.00	5/31/2023 7:37:28 am	107	1%
⋮	⋮	⋮	⋮	⋮
5/31/2023 11:56:27 am	18	5/31/2023 11:56:10 pm	18	0%
5/31/2023 4:43:45 pm	29	5/31/2023 4:43:21 pm	33	14%
5/31/2023 4:49:34 pm	31	5/31/2023 4:48:58 pm	40	29%
5/31/2023 4:53:30 pm	7	5/31/2023 4:53:23 pm	7	0%
⋮	⋮	⋮	⋮	⋮

After showing these results to the managers in the area, they acknowledged that the current replication is satisfactory. Although practical purposes have been achieved and there are no major issues, it cannot be claimed that a 100% R2 level has been attained since the recorded times do not align perfectly. Unfortunately, with the current sensing capabilities in this area, these minor differences between the physical and digital models cannot be fully eliminated. Therefore, this system falls somewhere between an R2 level DT and a data-generated simulation.

6.4.3 Reality (R3) and Relational (R4)

The goal of this research was to achieve as many levels of DT as possible. Implementing R1 and R2 was relatively straightforward using an existing DES. However, as mentioned above, some roadblocks were encountered getting exact replication at the R2 level. Furthermore, accomplishing R3 and R4 posed significant technological challenges that require further investigation. As a result, this research concluded after completing R1 and R2, with the potential for pursuing R3 and R4 as future research endeavors.

From the beginning, it was understood that task R4 could not be achieved using Simio, the chosen DES software, for this specific case study. Unfortunately, throughout the research process, specific technological limitations were encountered that restricted the scope to accomplishing only R1 and R2. It is important to note that Simio is widely recognized as one of the leading DES software solutions available. Nevertheless, at present, starting from DES software like Simio does not allow for progress beyond the R2 level, at least within the context of this case study. This limitation arises from the fact that Simio does not provide support for real-time interaction, it still requires a speed run approach, and it is unable to update tables while it is running. However, insightful discussions with Simio executives revealed that they are actively addressing these limitations. The upcoming release of Simio will incorporate modifications, such as the inclusion of real-time execution options instead of simply selecting the run speed. As a result, it is expected that within a few years, achieving R3 using Simio as a DES platform will be possible. R4, however, may require a solution with a more medium-term outlook.

6.5 Considerations and Drawbacks

In this specific system, it is important to acknowledge certain limitations regarding the connectivity and data retrieval processes. While automated connectivity is technically feasible, the automotive manufacturing company involved in this research has opted for manual data retrieval from the PLCs and cloud service using the designated software mentioned earlier. This manual data retrieval involves inputting the desired period and process into the software, which then generates an Excel file containing the requested data. It is important to highlight that despite this limitation, once the data is retrieved, the subsequent flow of data from the physical environment to the virtual environment occurs automatically. The collected data from the PLCs, sensors, and other sources is seamlessly transferred to the digital system. While this manual data retrieval approach may be suitable for the company’s current needs and preferences and may not be as time-consuming as initially described, it still involves certain limitations and considerations:

- Human involvement: The process still relies on human intervention to initiate and execute the data retrieval request. While the time required may not be significant, it still necessitates the availability of personnel to perform these tasks.
- Delayed data availability: Depending on the urgency of data requirements, there may be a potential delay between the time the request is made and when the data is received. This delay could impact real-time or time-sensitive analyses and decision-making processes.
- Potential for errors: Human error can still occur during the manual data retrieval process. Mistakes in inputting parameters or executing the request may lead to inaccurate or incomplete data being retrieved.

Another limitation encountered is related to the simulation software, Simio. It requires configuring the run speed, which does not precisely match real-time seconds and may run slightly faster or slower. Although this is an existing limitation of the software, efforts were made to address it by reaching out to the Simio software team. They have indicated that a new edition of Simio will be released, featuring a button to simulate in real-time, eliminating the need for manual run speed setup.

Another drawback that arises when striving for higher levels of DT, specifically R3 and R4, is the need to configure a starting time and end time at the beginning of each simulation run. This requirement poses a constraint as it hinders the seamless integration and continuous synchronization between the DT and the physical system. Achieving the desired levels of real-time, dynamic interaction and feedback between the DT and the physical system becomes challenging due to this setup limitation.

One crucial observation to highlight is that achieving Representation (R1) and Replication (R2) levels in a DT only required the first two levels of simulation capabilities: Modeling (S1) and Analyzing (S2). This leads to the conjecture that in general, DES models that attain capability levels S1 and S2 are likely to be suitable candidates for conversion into DTs that can achieve levels R1 and R2. However, it's important to note that the connection between these levels is not one-to-one. Furthermore, the question of how higher simulation capability levels, such as S3 and S4, will prove useful in creating DTs at R3 and R4 levels remains an open topic.

6.6 Generalized Approach for Transforming a DES to DT

In this research, a generalized approach for transforming DES to DT is proposed based on the steps undertaken in the case study. The aim of this approach is to provide a systematic and replicable method for converting a DES into a DT. However, it is important to note that the inductive approach used in this research limits the applicability of this generalized framework. We conjecture that this framework would easily apply to processes in assembly lines within the automotive manufacturing sector, but may not be directly applicable, for example, in a continuous manufacturing context. Despite this limitation, this approach remains valuable as it provides insights and a strong foundation for understanding the transformation from a DES to a DT. The steps involved in the transformation process, as demonstrated in the pallet elevator case study, serve as a guideline for organizations seeking to implement similar transformations. Additionally, this research can offer a starting point for future research and be a crucial stepping stone toward developing more comprehensive frameworks that encompass a broader range of industries and manufacturing processes.

6.6.1 Generalized Approach for Achieving Representation (R1) from a DES

The process of transforming a DES to achieve the R1 level of DT is illustrated in Table 57. It encompasses four main steps.

Table 57 General Steps for Achieving R1 from a DES.

	Achieving R1 from a DES
	1. Purpose and Scope Definition
	2. Data Mapping
	3. System Architecture Design
	4. Data Collection

<u>Step 1 – Purpose and Scope Definition</u>: The first step involves establishing a clear and precise purpose and scope of the transformation. This entails identifying the project's specific goals

and objectives, as well as outlining the boundaries and constraints of the system that will undergo the transition from a DES to a DT. Furthermore, it is crucial to identify the specific insights, enhancements, or capabilities that are sought through this transition. By clearly defining the purpose and scope, it becomes easier to align the transformation process with the desired outcomes.

Step 2 – Data Mapping: The next step is to understand the existing data used in the DES and map and classify the data elements that require modification or additional capture to effectively represent and integrate real-time data. This involves identifying the different types of data that are relevant to the system and determining whether they are static or dynamic in nature. Static data remains unchanged throughout the simulation, while dynamic data needs to be updated with real-time inputs. Additionally, it involves identifying the data sources required to collect real-time data, such as sensors or other monitoring devices. By mapping and classifying the data, it becomes possible to understand which elements are necessary for achieving the desired representation in the virtual environment.

Step 3 – System Architecture Design: The next step is to design the system architecture, which includes the physical environment, digital environment, cloud service, and digital interface. The physical environment involves the identification of the real-world components and processes that the DT aims to replicate. The digital environment involves identifying the virtual counterparts that will be utilized to represent the physical system. The cloud service component provides necessary infrastructure for storing and processing large amounts of data generated by the system. The digital interface involves establishing the means by which data will be collected from identified components, machines, the process, and the environment. This digital interface serves as the communication and interaction medium between the physical and digital components of the system. The system architecture should provide a structured and formal description of the system, highlighting its structure and behaviors. It should also consider the integration of sensors, applications, and digital interface tools to facilitate data collection and storage.

Step 4 – Data Collection: With the system architecture in place, the data collection plan can be implemented. This involves accessing the data sources, such as sensors or PLCs, and collecting the relevant data. During the data collection process, it is essential to ensure that the data formats obtained from the sources are compatible with the intended data analysis software or tools. This compatibility facilitates seamless processing and interpretation of the collected data. Additionally, adequate computer storage capacity should be available to accommodate the collected data, ensuring that it can be effectively managed and utilized throughout the DT implementation.

6.6.2 Generalized Approach for Achieving Replication (R2) from a DES

The process of transforming a DES to achieve the R2 level of DT is illustrated in Table 58. It encompass three main steps.

Table 58 General Steps for Achieving R2 from a DES.

	Achieving R2 from a DES
	1. Map the Virtual Environment Elements
	2. Integrate the Simulation Model and Data Sources
	3. Verification and Validation of the DT

Step 1 – Map the Virtual Environment Elements: The initial step involves evaluating and mapping the current elements present in the virtual system of the DES to determine their suitability in representing the desired aspects of the DT. If the mapping process indicates that the existing elements in the virtual system align with the key variables and characteristics of the desired system, it indicates that the DES can serve as a solid foundation for the representation of the DT, requiring no additional modifications or components. However, in most cases, certain changes may be necessary to enable the required functionality for replicating and representing the behavior and performance of the physical system within the virtual environment. Regardless, the existing DES can be leveraged as a dependable basis for constructing and operating the DT, facilitating a smooth transition from the physical system to the virtual system.

Step 2 – Integrate the Simulation Models and Data Sources: This step focuses on the integration of the simulation model with real-time data sources. It entails incorporating the identified data types from the previous level, which is the representation phase, into the virtual system to replicate the behavior of the physical system. During this integration process, the data that was defined in the data architecture phase is incorporated into the simulation model. This integration involves establishing connections and interfaces between the simulation model and the real-time data sources. It enables the flow of data from the physical system to the virtual environment, allowing the DT to interact and respond to real-time inputs and events. This integration enables the DT to incorporate dynamic and up-to-date information, enhancing its ability to simulate and predict real-world scenarios accurately.

Step 3 – Verification and Validation of the DT: In this step, the verification and validation of the transformed DT occurs by comparing its behavior and performance to that of the real system.

The main goal is to ensure that the DT generates identical outputs when presented with the same inputs as the physical system, and accurately represents the behavior of the real system. This involves inputting specific conditions or events to observe and analyze how the DT responds. By carefully carrying out verification and validation activities, any discrepancies between the DT and the physical system can be identified and resolved. This step is crucial for ensuring the reliability and fidelity of the DT, instilling confidence in its ability to simulate and predict real-world scenarios.

Given that the case study presented did not achieve the R3 and R4 levels of the DT, this research does not provide a generalized approach for these levels. The absence of a generalized approach implies that further research and development are necessary to explore and define techniques specific to attaining R3 and R4 levels from a DES. It highlights the need for future investigations and experimentation to unlock the full potential of using DES as the basis for DT in relation to these advanced levels of DT capabilities.

6.7 Chapter Summary

This research aimed to explore and document the process of transforming a fully capable DES into a DT using the 4S and 4R frameworks. This research contributes to a clearer understanding of DT application by offering two main contributions:

- Identification and summarization of the key steps involved in the transformation from a DES to a DT in a general and comprehensive manner.
- Practical demonstration of bridging the gap between DES and DT, fully achieving R1 level and partially achieving R2 level, highlighting the potential benefits for the advancement of simulation and modeling.

Overall, this research provides valuable insights into the transformation process and demonstrates the potential advantages of adopting DT approaches in simulation and modeling. By bridging the gap between simulation and DT, this research endeavors to unlock new possibilities for system modeling, analysis, and optimization. The systematic transformation of simulations into DTs will facilitate real-time monitoring, advanced analytics, and decision-making based on accurate representations of physical systems. This research aims to contribute to the advancement of simulation and modeling practices, enabling the widespread adoption of DT technology across various industries.

Additionally, it's worth noting that reaching the Representation (R1) and Replication (R2) in a DT only required the first two levels of simulation capabilities: Modeling (S1) and Analyzing (S2). The existing knowledge gained from the DES during the S1 and S2 levels served as a foundation for developing the DT, providing essential information about the system’s behavior. Therefore, it can be conjectured, that in general, DES models that achieve capability levels S1 and S2 are suitable for conversion to DT that achieve levels R1 and R2. Note however, that it is not a one-to-one connection, to achieve R1, a simulation that achieved level S2 was necessary. The question of whether capability levels S3 and S4 will be useful for creating DTs at levels R3 and R4 remains open.

In summary, the existing understanding of the physical system's behavior, acquired through the development of the DES, plays a crucial role in the transformation to a DT. It provides a solid foundation for developing a more advanced and dynamic simulation model that incorporates real-time data, enabling a higher level of accuracy and responsiveness in the DT.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This research unveiled the distinctions between simulation and DT models, emphasizing that they should be recognized as separate technologies with unique capabilities. It enhanced the understanding of these domains for academia and industry, facilitating the pursuit of more impactful applications and research. This research also underscored the potential of simulation as a precursor to DT development, allowing researchers to leverage existing simulation capabilities and expand them to encompass DT capabilities.

The first contribution of this research was the development and application of the 4S framework, which defines the levels of capability for simulation models. This framework demonstrated how to progressively enhance the capability of a simulation model, one level at a time. Through a successful simulation application based on the 4S framework, the effectiveness of this approach was demonstrated. This contribution addressed a gap in the existing literature by providing a clear understanding of the capability levels of simulation models that can be directly compared to the capability levels of DTs. Prior to this research, there was a lack of direct comparison and understanding in this regard. Additionally, existing frameworks did not offer a clear demonstration of how to sequentially add capabilities to a simulation model.

The second contribution was a systematic literature review to investigate the application of simulation-based DT in journal publications. By utilizing the 4R and 4S frameworks, this research classified the works based on their capabilities, allowing for a clear distinction between simulation and DT. This analysis helped to address the confusion that arises when simulation models are misclassified as DT and vice versa. The impact of this contribution is twofold. First, it enhances the knowledge and awareness of researchers, practitioners, and decision-makers about the proper classification and differentiation of simulation models and DTs. This understanding is crucial for effectively utilizing and implementing these technologies in various domains. Second, by addressing the confusion and providing clear distinctions, this research enables researchers and practitioners to make informed decisions about the appropriate tools and techniques to employ in their specific contexts. It helps in avoiding misinterpretations of simulation models as DTs.

The third contribution aimed to bridge the gap between simulation and DT by providing a practical guideline for researchers to transform a fully capable simulation into a DT. By demonstrating the specific steps involved in this transformation process, this research demonstrated how simulations can serve as a foundation for building DTs. The guideline provided researchers with a clear roadmap to follow, enabling them to bridge the gap and leverage the advantages of both simulation and DT technologies. This contribution provides practical assistance in translating the capabilities of a simulation model into the context of a DT, allowing researchers and manufacturers to explore the benefits of DTs without starting from scratch. Additionally, it's worth noting that reaching the Representation (R1) and Replication (R2) in a DT only required the first two levels of simulation capabilities: Modeling (S1) and Analyzing (S2). The existing knowledge gained from the DES during the S1 and S2 levels served as a foundation for developing the DT, providing essential information about the system’s behavior.

Overall, this research has significant implications for academia and industry. It provides a clearer understanding of the differences in capabilities between simulation and DT, helping to prevent misunderstandings and ensure the appropriate classification of models. The findings of this research contribute to achieving consistency in DT implementations and enable more effective and relevant applications in the future.

7.2 Future Work

Future research should expand upon the guideline developed in this study to transform DES into DT with the focus on developing higher levels of DT capabilities. This involves identifying and incorporating additional features and functionalities into the transformed simulation model to enhance its accuracy and utility as a DT. Advanced data integration and analysis techniques can be explored to enable real-time data processing and decision-making within the DT. This would contribute to a more dynamic and responsive virtual representation of the physical system.

Furthermore, future research should investigate the impact of DT implementation on various industrial sectors. By conducting case studies that deploy DTs in real production environments, researchers can assess the performance and effectiveness of the DTs and identify areas for improvement. This would provide valuable insights into the practical implications and benefits of DTs in different industry settings.

To advance the understanding and utilization of both simulation and DT, future research should also focus on refining the capability frameworks introduced in this study. This may involve incorporating additional levels of capability or refining the existing levels to better align with the evolving needs of DT implementations.

Exploring the integration of advanced technologies with DTs is another important avenue for future research. This includes investigating how emerging technologies such as AI, ML, IoT, and big data analytics can be integrated into DTs to enhance their capabilities and provide more comprehensive insights. Additionally, efforts should be made to establish standardization efforts and best practices for DT implementation, promoting interoperability and consistency across different applications and industries.

By addressing these research directions, the academic and industrial communities can advance the understanding and utilization of both simulation and DT. This will foster the development of more robust and accurate models, enabling researchers and practitioners to harness the full potential of these technologies and drive innovation in various domains.

References

[1] Roblek V, Meško M, Krapež A. A complex view of industry 4.0. Sage Open 2016;6:2158244016653987.

[2] Schwab K. The fourth industrial revolution. Currency; 2017.

[3] Haag S, Anderl R. Digital twin–Proof of concept. Manuf Lett 2018;15:64–6.

[4] Negri E, Fumagalli L, Macchi M. A review of the roles of digital twin in CPS-based production systems. Procedia Manuf 2017;11:939–48.

[5] Grieves M. Digital Twin: Manufacturing Excellence through Virtual Factory Replication. 2014.

[6] Tao F, Cheng J, Qi Q, Zhang M, Zhang H, Sui F. Digital twin-driven product design, manufacturing and service with big data. International Journal of Advanced Manufacturing Technology 2018;94:3563–76. https://doi.org/10.1007/s00170-017-0233-1.

[7] Shao G, Helu M. Framework for a digital twin in manufacturing: Scope and requirements. Manuf Lett 2020;24:105–7.

[8] Banks J. Principles of simulation. Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice 1998:3–30.

[9] Negahban A, Smith JS. Simulation for manufacturing system design and operation: Literature review and analysis. J Manuf Syst 2014;33:241–61.

[10] Boschert S, Rosen R. Digital twin—the simulation aspect. Mechatronic futures, Springer; 2016, p. 59–74.

[11] Jahangirian M, Eldabi T, Naseer A, Stergioulas LK, Young T. Simulation in manufacturing and business: A review. Eur J Oper Res 2010;203:1–13.

[12] Schriber TJ, Brunner DT. Inside discrete-event simulation software: How it works and why it matters. Proceedings of the 29th conference on Winter simulation, 1997, p. 14–22.

[13] Kritzinger W, Karner M, Traar G, Henjes J, Sihn W. Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine, vol. 51, Elsevier B.V.; 2018, p. 1016–22. https://doi.org/10.1016/j.ifacol.2018.08.474.

[14] Wright L, Davidson S. How to tell the difference between a model and a digital twin. Adv Model Simul Eng Sci 2020;7. https://doi.org/10.1186/s40323-020-00147-4.

[15] van der Valk H, Hunker J, Rabe M, Otto B. Digital twins in simulative applications: A taxonomy. 2020 Winter Simulation Conference (WSC), IEEE; 2020, p. 2695–706.

[16] Liu M, Fang S, Dong H, Xu C. Review of digital twin about concepts, technologies, and industrial applications. J Manuf Syst 2021;58:346–61. https://doi.org/10.1016/j.jmsy.2020.06.017.

[17] Lasi H, Fettke P, Kemper H-G, Feld T, Hoffmann M. Industry 4.0. Business & Information Systems Engineering 2014;6:239–42. https://doi.org/10.1007/s12599-014-0334-4.

[18] Xu L Da, Xu EL, Li L. Industry 4.0: state of the art and future trends. Int J Prod Res 2018;56:2941–62. https://doi.org/10.1080/00207543.2018.1444806.

[19] Lasa IS, Laburu CO, de Castro Vila R. An evaluation of the value stream mapping tool. Business Process Management Journal 2008.

[20] Rother M, Shook J. Learning to see: value stream mapping to add value and eliminate muda. Lean enterprise institute; 2003.

[21] Donatelli AJ, Harris GA. Combining value stream mapping and discrete event simulation. Proceedings of the Huntsville Simulation Conference,[online, 21.02. 2012]: https://rsic. redstone. army. mil/hsc/papers/hsc003. pdf, 2001.

[22] Andrade PF, Pereira VG, Del Conte EG. Value stream mapping and lean simulation: a case study in automotive company. The International Journal of Advanced Manufacturing Technology 2016;85:547–55.

[23] Moreira ACM, Silva DF. Applying Discrete-Event Simulation and Value Stream Mapping to Reduce Waste in an Automotive Engine Manufacturing Plant. 2021 Winter Simulation Conference (WSC), 2021, p. 1–12. https://doi.org/10.1109/WSC52266.2021.9715477.

[24] McHaney R. Computer simulation: a practical perspective. Academic press; 1991.

[25] Lins T, Oliveira RAR, Correia LHA, Silva JMS. Industry 4.0 Retrofitting. 2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC) 2018:8–15.

[26] Guerra-Zubiaga D, Kuts V, Mahmood K, Bondar A, Nasajpour-Esfahani N, Otto T. An approach to develop a digital twin for industry 4.0 systems: manufacturing automation case studies. Int J Comput Integr Manuf 2021;34:933–49. https://doi.org/10.1080/0951192X.2021.1946857.

[27] Diez-Olivan A, del Ser J, Galar D, Sierra B. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. Information Fusion 2019;50:92–111.

[28] Yu W, Liu Y, Dillon T, Rahayu W, Mostafa F. An integrated framework for health state monitoring in a smart factory employing IoT and big data techniques. IEEE Internet Things J 2021;9:2443–54.

[29] Perno M, Hvam L, Haug A. Implementation of digital twins in the process industry: A systematic literature review of enablers and barriers. Comput Ind 2022;134. https://doi.org/10.1016/j.compind.2021.103558.

[30] Barrera-Diaz CA, Oscarsson J, Lidberg S, Sellgren T. Discrete event simulation output data-handling system in an automotive manufacturing plant. Procedia Manuf 2018;25:23–30.

[31] Lian Y-H, van Landeghem H. An application of simulation and value stream mapping in lean manufacturing. Proceedings 14th European Simulation Symposium, c) SCS Europe BVBA; 2002, p. 1–8.

[32] Mahfouz A, Arisha A. Lean distribution assessment using an integrated framework of value stream mapping and simulation. 2013 Winter Simulations Conference (WSC), IEEE; 2013, p. 3440–9.

[33] Antonelli D, Stadnicka D. Combining factory simulation with value stream mapping: a critical discussion. Procedia CIRP 2018;67:30–5.

[34] Morgan CB. Discrete-event system simulation 1984.

[35] Beck A. Simulation: the practice of model development and use. Journal of Simulation 2008;2:67. https://doi.org/10.1057/palgrave.jos.4250031.

[36] Yin C, McKay A. Introduction to Modeling and Simulation Techniques. 2018.

[37] Law AM, Kelton WD, Kelton WD. Simulation modeling and analysis. vol. 3. Mcgraw-hill New York; 2007.

[38] Varga A. Discrete event simulation system. Proc. of the European Simulation Multiconference (ESM'2001), 2001, p. 1–7.

[39] Smith JS, Sturrock DT. Simio and Simulation - Modeling, Analysis, Applications . 6th Edition. 2021.

[40] Maria A. Introduction to modeling and simulation. Proceedings of the 29th conference on Winter simulation, 1997, p. 7–13.

[41] Fishman GS. Principles of discrete event simulation.[book review] 1978.

[42] Shanton K, Goldman A. Simulation theory. Wiley Interdiscip Rev Cogn Sci 2010;1:527–38.

[43] Schluse M, Rossmann J. From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems. ISSE 2016 - 2016 International Symposium on Systems Engineering - Proceedings Papers, Institute of Electrical and Electronics Engineers Inc.; 2016. https://doi.org/10.1109/SysEng.2016.7753162.

[44] MOSIMTEC. 4 Types of Simulation Models to Leverage in Your Business 2022. https://mosimtec.com/types-of-simulation-models/ (accessed December 17, 2022).

[45] Harrison RL. Introduction to monte carlo simulation. AIP Conf Proc, vol. 1204, American Institute of Physics; 2010, p. 17–21.

[46] Davidsson P, Holmgren J, Kyhlbäck H, Mengistu D, Persson M. Applications of agent based simulation. International Workshop on Multi-Agent Systems and Agent-Based Simulation, Springer; 2006, p. 15–27.

[47] Banks J. Introduction to simulation. Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future-Volume 1, 1999, p. 7–13.

[48] Greasley A, editor. Introduction to Simulation. Enabling a Simulation Capability in the Organisation, London: Springer London; 2008, p. 1–6. https://doi.org/10.1007/978-1-84800-169-5_1.

[49] Ingalls RG. Introduction to simulation. Proceedings of the 2011 Winter Simulation Conference (WSC), 2011, p. 1374–88. https://doi.org/10.1109/WSC.2011.6147858.

[50] Bennett BS. Simulation fundamentals. Prentice Hall International (UK) Ltd.; 1995.

[51] Kuhn W. Digital factory-simulation enhancing the product and production engineering process. Proceedings of the 2006 winter simulation conference, IEEE; 2006, p. 1899–906.

[52] Cassandras CG, Lafortune S, editors. Sensitivity Analysis and Concurrent Estimation. Introduction to Discrete Event Systems, Boston, MA: Springer US; 2008, p. 617–740. https://doi.org/10.1007/978-0-387-68612-7_11.

[53] Anglani A, Grieco A, Pacella M, Tolio T. Object-oriented modeling and simulation of flexible manufacturing systems: a rule-based procedure. Simul Model Pract Theory 2002;10:209–34.

[54] Robinson S. Simulation: the practice of model development and use. Bloomsbury Publishing; 2014.

[55] Pidd M. Computer modelling for discrete simulation. John Wiley & Sons, Inc.; 1989.

[56] Banks J. Discrete event system simulation. Pearson Education India; 2005.

[57] Law AM, McComas MG. Secrets of successful simulation studies. Institute of Electrical and Electronics Engineers (IEEE); 1991.

[58] Babulak E, Wang M. Discrete event simulation. Aitor Goti (Hg): Discrete Event Simulations Rijeka, Kroatien: Sciyo 2010:1.

[59] Williams EJ, Ülgen OM. Simulation applications in the automotive industry. Use Cases of Discrete Event Simulation, Springer; 2012, p. 45–58.

[60] Renteria-Marquez IA, Almeraz CN, Tseng T-LB, Renteria A. A heijunka study for automotive assembly using discrete-event simulation: a case study. 2020 Winter Simulation Conference (WSC), IEEE; 2020, p. 1641–51.

[61] Alsakka F, Khalife S, Darwish M, Al-Hussein M, Mohamed Y. Deploying discrete-event simulation and continuous improvement to increase production rate in a modular construction facility. 2020 Winter Simulation Conference (WSC), IEEE; 2020, p. 1676–87.

[62] Wang J, Chang Q, Xiao G, Wang N, Li S. Data driven production modeling and simulation of complex automobile general assembly plant. Comput Ind 2011;62:765–75.

[63] Michalos G, Makris S, Papakostas N, Mourtzis D, Chryssolouris G. Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach. CIRP J Manuf Sci Technol 2010;2:81–91.

[64] Qi Y, Mao Z, Zhang M, Guo H. Manufacturing practices and servitization: The role of mass customization and product innovation capabilities. Int J Prod Econ 2020;228:107747.

[65] Brooks RA. Intelligence without representation. Artif Intell 1991;47:139–59.

[66] Balci O. Verification validation and accreditation of simulation models. Proceedings of the 29th conference on Winter simulation, 1997, p. 135–41.

[67] Lazarova-Molnar S, Li X. Deriving simulation models from data: steps of simulation studies revisited. 2019 Winter Simulation Conference (WSC), IEEE; 2019, p. 2771–82.

[68] Borshchev A. Multi-method modelling: AnyLogic. Discrete-Event Simulation and System Dynamics for Management Decision Making 2014:248–79.

[69] Nuruzzaman M. Modeling and Simulation in SIMULINK for Engineers and Scientists. AuthorHouse; 2005.

[70] Lattner AD, Bogon T, Lorion Y, Timm IJ. A knowledge-based approach to automated simulation model adaptation. Proceedings of the 2010 Spring Simulation Multiconference, 2010, p. 1–8.

[71] Wang X, Wang Y, Tao F, Liu A. New paradigm of data-driven smart customisation through digital twin. J Manuf Syst 2021;58:270–80.

[72] Sarker IH. AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems. SN Comput Sci 2022;3:158. https://doi.org/10.1007/s42979-022-01043-x.

[73] Botín-Sanabria DM, Mihaita A-S, Peimbert-García RE, Ramírez-Moreno MA, Ramírez-Mendoza RA, Lozoya-Santos J de J. Digital Twin Technology Challenges and Applications: A Comprehensive Review. Remote Sens (Basel) 2022;14. https://doi.org/10.3390/rs14061335.

[74] Ludwig S, Karrenbauer M, Fellan A, Schotten HD, Buhr H, Seetaraman S, et al. A 5G Architecture for The Factory of the Future 2018.

[75] Sharma A, Kosasih E, Zhang J, Brintrup A, Calinescu A. Digital Twins: State of the art theory and practice, challenges, and open research questions. J Ind Inf Integr 2022:100383. https://doi.org/10.1016/j.jii.2022.100383.

[76] Syafrudin M, Alfian G, Fitriyani NL, Rhee J. Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing. Sensors (Switzerland) 2018;18. https://doi.org/10.3390/s18092946.

[77] Tao F, Zhang H, Liu A, Nee AYC. Digital Twin in Industry: State-of-the-Art. IEEE Trans Industr Inform 2019;15:2405–15. https://doi.org/10.1109/TII.2018.2873186.

[78] Melesse TY, di Pasquale V, Riemma S. Digital Twin models in industrial operations: State-of-the-art and future research directions. IET Collaborative Intelligent Manufacturing 2021;3:37–47.

[79] VanDerHorn E, Mahadevan S. Digital Twin: Generalization, characterization and implementation. Decis Support Syst 2021;145:113524. https://doi.org/https://doi.org/10.1016/j.dss.2021.113524.

[80] Walton RB, Ciarallo FW. A Framework to Manage Systems Based on Digital Twin. IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE); 2022, p. 1–6.

[81] Jones D, Snider C, Nassehi A, Yon J, Hicks B. Characterising the Digital Twin: A systematic literature review. CIRP J Manuf Sci Technol 2020;29:36–52. https://doi.org/10.1016/j.cirpj.2020.02.002.

[82] Boschert S, Heinrich C, Rosen R. Next generation digital twin. Proc. tmce, vol. 2018, Las Palmas de Gran Canaria, Spain; 2018, p. 7–11.

[83] Grieves M, Vickers J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: Kahlen F-J, Flumerfelt S, Alves A, editors. Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches, Cham: Springer International Publishing; 2017, p. 85–113. https://doi.org/10.1007/978-3-319-38756-7_4.

[84] Abramovici M, Göbel JC, Savarino P. Virtual Twins as Integrative Components of Smart Products. In: Harik R, Rivest L, Bernard A, Eynard B, Bouras A, editors. Product Lifecycle Management for Digital Transformation of Industries, Cham: Springer International Publishing; 2016, p. 217–26.

[85] Schroeder GN, Steinmetz C, Pereira CE, Espindola DB. Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange. IFAC-PapersOnLine 2016;49:12–7. https://doi.org/https://doi.org/10.1016/j.ifacol.2016.11.115.

[86] Shao G, Kibira D. Digital manufacturing: Requirements and challenges for implementing digital surrogates. 2018 Winter Simulation Conference (WSC), IEEE; 2018, p. 1226–37.

[87] Tao F, Zhang M. Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. IEEE Access 2017;5:20418–27. https://doi.org/10.1109/ACCESS.2017.2756069.

[88] Gabor T, Belzner L, Kiermeier M, Beck MT, Neitz A. A Simulation-Based Architecture for Smart Cyber-Physical Systems. 2016 IEEE International Conference on Autonomic Computing (ICAC) 2016:374–9.

[89] Rosen R, von Wichert G, Lo G, Bettenhausen KD. About The Importance of Autonomy and Digital Twins for the Future of Manufacturing. IFAC-PapersOnLine 2015;48:567–72. https://doi.org/https://doi.org/10.1016/j.ifacol.2015.06.141.

[90] Coronado PDU, Lynn R, Louhichi W, Parto M, Wescoat E, Kurfess T. Part data integration in the Shop Floor Digital Twin: Mobile and cloud technologies to enable a manufacturing execution system. J Manuf Syst 2018;48:25–33.

[91] Singh M, Fuenmayor E, Hinchy EP, Qiao Y, Murray N, Devine D. Digital twin: Origin to future. Applied System Innovation 2021;4. https://doi.org/10.3390/asi4020036.

[92] Juarez MG, Botti VJ, Giret AS. Digital twins: Review and challenges. J Comput Inf Sci Eng 2021;21. https://doi.org/10.1115/1.4050244.

[93] Lim KYH, Zheng P, Chen CH. A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives. J Intell Manuf 2020;31:1313–37. https://doi.org/10.1007/s10845-019-01512-w.

[94] Rathore MM, Shah SA, Shukla D, Bentafat E, Bakiras S. The Role of AI, Machine Learning, and Big Data in Digital Twinning: A Systematic Literature Review, Challenges, and Opportunities. IEEE Access 2021;9:32030–52. https://doi.org/10.1109/ACCESS.2021.3060863.

[95] Barricelli BR, Casiraghi E, Fogli D. A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. IEEE Access 2019;7:167653–71. https://doi.org/10.1109/ACCESS.2019.2953499.

[96] Hyre A, Harris G, Osho J, Pantelidakis M, Mykoniatis K, Liu J. Digital twins: Representation, Replication, Reality, and Relational (4Rs). Manuf Lett 2022;31:20–3. https://doi.org/10.1016/j.mfglet.2021.12.004.

[97] Osho J, Hyre A, Pantelidakis M, Ledford A, Harris G, Liu J, et al. Four Rs Framework for the development of a digital twin: The implementation of Representation with a FDM manufacturing machine. J Manuf Syst 2022;63:370–80. https://doi.org/10.1016/j.jmsy.2022.04.014.

[98] Ante L. Digital twin technology for smart manufacturing and industry 4.0: A bibliometric analysis of the intellectual structure of the research discourse. Manuf Lett 2021;27:96–102. https://doi.org/10.1016/j.mfglet.2021.01.003.

[99] Chung CA. Simulation modeling handbook: a practical approach. CRC press; 2003.

[100] Sokolowski JA, Banks CM. Principles of modeling and simulation: a multidisciplinary approach. John Wiley & Sons; 2011.

[101] Bossel H. Modeling and simulation. Springer-Verlag; 2013.

[102] Bandyopadhyay S, Bhattacharya R. Discrete and continuous simulation: theory and practice. CRC Press; 2014.

[103] Shannon RE. Introduction to the art and science of simulation. 1998 winter simulation conference. proceedings (cat. no. 98ch36274), vol. 1, IEEE; 1998, p. 7–14.

[104] Carson JS. Introduction to modeling and simulation. Proceedings of the Winter Simulation Conference, 2005., IEEE; 2005, p. 8-pp.

[105] Goldsman D. Introduction to simulation. 2007 Winter Simulation Conference, IEEE; 2007, p. 26–37.

[106] Sanchez PJ. As simple as possible, but no simpler: a gentle introduction to simulation modeling. Proceedings of the 2006 winter simulation conference, IEEE; 2006, p. 2–10.

[107] White KP, Ingalls RG. The basics of simulation. 2018 Winter Simulation Conference (WSC), IEEE; 2018, p. 147–61.

[108] Robinson S, Arbez G, Birta LG, Tolk A, Wagner G. Conceptual modeling: Definition, purpose and benefits. 2015 Winter Simulation Conference (WSC), 2015, p. 2812–26. https://doi.org/10.1109/WSC.2015.7408386.

[109] Davenport TH, Harris JG, Morison R. Analytics at work: Smarter decisions, better results. Harvard Business Press; 2010.

[110] Oberkampf WL, Trucano TG, Pilch MM. Predictive Capability Maturity Model for computational modeling and simulation. Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA …; 2007.

[111] Sargent RG. Verification and validation of simulation models. Proceedings of the 2010 winter simulation conference, IEEE; 2010, p. 166–83.

[112] von Rueden L, Mayer S, Sifa R, Bauckhage C, Garcke J. Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions. International Symposium on Intelligent Data Analysis, 2020.

[113] Sturrock DT, Pegden CD. Recent innovations in Simio. Proceedings of the 2011 Winter Simulation Conference (WSC), IEEE; 2011, p. 52–62.

[114] Adacher L, Flamini M. Optimizing airport land side operations: check-in, passengers' migration, and security control processes. J Adv Transp 2020;2020:1–14.

[115] Laguna M. OptQuest. Optimization of Complex Systems 2011.

[116] Grieves M. Digital Twin: Manufacturing Excellence through Virtual Factory Replication 2015.

[117] Opoku DGJ, Perera S, Osei-Kyei R, Rashidi M. Digital twin application in the construction industry: A literature review. Journal of Building Engineering 2021;40. https://doi.org/10.1016/j.jobe.2021.102726.

[118] Opoku DGJ, Perera S, Osei-Kyei R, Rashidi M, Famakinwa T, Bamdad K. Drivers for Digital Twin Adoption in the Construction Industry: A Systematic Literature Review. Buildings 2022;12. https://doi.org/10.3390/buildings12020113.

[119] Nguyen T, Duong QH, Nguyen T van, Zhu Y, Zhou L. Knowledge mapping of digital twin and physical internet in Supply Chain Management: A systematic literature review. Int J Prod Econ 2022;244. https://doi.org/10.1016/j.ijpe.2021.108381.

[120] Lo CK, Chen CH, Zhong RY. A review of digital twin in product design and development. Advanced Engineering Informatics 2021;48. https://doi.org/10.1016/j.aei.2021.101297.

[121] Botín-Sanabria DM, Mihaita S, Peimbert-García RE, Ramírez-Moreno MA, Ramírez-Mendoza RA, Lozoya-Santos J de J. Digital Twin Technology Challenges and Applications: A Comprehensive Review. Remote Sens (Basel) 2022;14. https://doi.org/10.3390/rs14061335.

[122] Davila Delgado JM, Oyedele L. Digital Twins for the built environment: learning from conceptual and process models in manufacturing. Advanced Engineering Informatics 2021;49. https://doi.org/10.1016/j.aei.2021.101332.

[123] Briner RB, Denyer D. Systematic Review and Evidence Synthesis as a Practice and Scholarship Tool. The Oxford Handbook of Evidence-Based Management, Oxford University Press; 2012. https://doi.org/10.1093/oxfordhb/9780199763986.013.0007.

[124] Zhang Z, Guan Z, Gong Y, Luo D, Yue L. Improved multi-fidelity simulation-based optimisation: application in a digital twin shop floor. Int J Prod Res 2022;60:1016–35. https://doi.org/10.1080/00207543.2020.1849846.

[125] Guo J, Wu Q, Guo K, Xiong S, Feng W, Xue J. Study on the Construction and Application of Digital Twins on High Voltage Transmission Line Live Working Scenes. IEEE Access 2021;9:111587–94. https://doi.org/10.1109/ACCESS.2021.3097179.

[126] Wang G, Cao Y, Zhang Y. Digital twin-driven clamping force control for thin-walled parts. Advanced Engineering Informatics 2022;51. https://doi.org/10.1016/j.aei.2021.101468.

[127] Vachálek J, Šišmišová D, Vašek P, Fit'ka I, Slovák J, Šimovec M. Design and implementation of universal cyber-physical model for testing logistic control algorithms of production line's digital twin by using color sensor. Sensors 2021;21:1–12. https://doi.org/10.3390/s21051842.

[128] Marmolejo-Saucedo JA. Design and Development of Digital Twins: a Case Study in Supply Chains. Mobile Networks and Applications 2020;25:2141–60. https://doi.org/10.1007/s11036-020-01557-9.

[129] Leng J, Yan D, Liu Q, Zhang H, Zhao G, Wei L, et al. Digital twin-driven joint optimisation of packing and storage assignment in large-scale automated high-rise warehouse product-service system. Int J Comput Integr Manuf 2021;34:783–800. https://doi.org/10.1080/0951192X.2019.1667032.

[130] Xu Z, Ji F, Ding S, Zhao Y, Zhou Y, Zhang Q, et al. Digital twin-driven optimization of gas exchange system of 2-stroke heavy fuel aircraft engine. J Manuf Syst 2021;58:132–45. https://doi.org/10.1016/j.jmsy.2020.08.002.

[131] Wong EYC, Mo DY, So S. Closed-loop digital twin system for air cargo load planning operations. Int J Comput Integr Manuf 2021;34:801–13. https://doi.org/10.1080/0951192X.2020.1775299.

[132] Leng J, Liu Q, Ye S, Jing J, Wang Y, Zhang C, et al. Digital twin-driven rapid reconfiguration of the automated manufacturing system via an open architecture model. Robot Comput Integr Manuf 2020;63. https://doi.org/10.1016/j.rcim.2019.101895.

[133] Lei L, Shen G, Zhang L, Li Z. Toward Intelligent Cooperation of UAV Swarms: When Machine Learning Meets Digital Twin. IEEE Netw 2021;35:386–92. https://doi.org/10.1109/MNET.011.2000388.

[134] Jiang H, Qu T, Wan M, Tang L, Huang GQ. Digital-twin-based implementation framework of production service system for highly dynamic production logistics operation. IET Collaborative Intelligent Manufacturing 2020;2. https://doi.org/10.1049/iet-cim.2019.0065.

[135] Wang Y, Ren W, Li Y, Zhang C. Complex product manufacturing and operation and maintenance integration based on digital twin. The International Journal of Advanced Manufacturing Technology 2021;117:361–81. https://doi.org/10.1007/s00170-021-07350-6.

[136] Yujun L, Zhichang Z, Wei W, Kui Z. Digital twin product lifecycle system dedicated to the constant velocity joint. Computers & Electrical Engineering 2021;93:107264. https://doi.org/10.1016/j.compeleceng.2021.107264.

[137] Xu LZ, Xie QS. Dynamic production scheduling of digital twin job-shop based on edge computing. Journal of Information Science and Engineering 2021;37:93–105. https://doi.org/10.6688/JISE.202101_37(1).0007.

[138] Zhao L, Fang Y, Lou P, Yan J, Xiao A. Cutting Parameter Optimization for Reducing Carbon Emissions Using Digital Twin. International Journal of Precision Engineering and Manufacturing 2021;22:933–49. https://doi.org/10.1007/s12541-021-00486-1.

[139] Xu W, Cui J, Li L, Yao B, Tian S, Zhou Z. Digital twin-based industrial cloud robotics: Framework, control approach and implementation. J Manuf Syst 2021;58:196–209. https://doi.org/10.1016/j.jmsy.2020.07.013.

[140] Glatt M, Sinnwell C, Yi L, Donohoe S, Ravani B, Aurich JC. Modeling and implementation of a digital twin of material flows based on physics simulation. J Manuf Syst 2021;58:231–45. https://doi.org/10.1016/j.jmsy.2020.04.015.

[141] Jiang H, Qu T, Wan M, Tang L, Huang GQ. A digital-twin-based implementation framework of production service system for highly dynamic production logistics operation. n.d.

[142] Wang Y, Ren W, Li Y, Zhang C. Complex product manufacturing and operation and maintenance integration based on digital twin n.d. https://doi.org/10.1007/s00170-021-07350-6/Published.

[143] Bilberg A, Malik AA. Digital twin driven human–robot collaborative assembly. CIRP Annals 2019;68:499–502. https://doi.org/10.1016/j.cirp.2019.04.011.

[144] Santos CH dos, de Queiroz JA, Leal F, Montevechi JAB. Use of simulation in the industry 4.0 context: Creation of a Digital Twin to optimise decision making on non-automated process. Journal of Simulation 2022;16:284–97. https://doi.org/10.1080/17477778.2020.1811172.

[145] Venkatesan S, Manickavasagam K, Tengenkai N, Vijayalakshmi N. Health monitoring and prognosis of electric vehicle motor using intelligent-digital twin. IET Electr Power Appl 2019;13:1328–35. https://doi.org/10.1049/iet-epa.2018.5732.

[146] Du J, Zhu Q, Shi Y, Wang Q, Lin Y, Zhao D. Cognition Digital Twins for Personalized Information Systems of Smart Cities: Proof of Concept. Journal of Management in Engineering 2020;36. https://doi.org/10.1061/(asce)me.1943-5479.0000740.

[147] Pengnoo M, Barros MT, Wuttisittikulkij L, Butler B, Davy A, Balasubramaniam S. Digital Twin for Metasurface Reflector Management in 6G Terahertz Communications. IEEE Access 2020;8:114580–96. https://doi.org/10.1109/ACCESS.2020.3003734.

[148] Bambura R, Šolc M, Dado M, Kotek L. Implementation of digital twin for engine block manufacturing processes. Applied Sciences (Switzerland) 2020;10. https://doi.org/10.3390/APP10186578.

[149] Liu Q, Leng J, Yan D, Zhang D, Wei L, Yu A, et al. Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system. J Manuf Syst 2021;58:52–64. https://doi.org/10.1016/j.jmsy.2020.04.012.

[150] Ma J, Chen H, Zhang Y, Guo H, Ren Y, Mo R, et al. A digital twin-driven production management system for production workshop. The International Journal of Advanced

Manufacturing Technology 2020;110:1385–97. https://doi.org/10.1007/s00170-020-05977-5.

[151] Zhang M, Tao F, Nee AYC. Digital Twin Enhanced Dynamic Job-Shop Scheduling. J Manuf Syst 2021;58:146–56. https://doi.org/10.1016/j.jmsy.2020.04.008.

[152] Lee D, Lee S. Digital twin for supply chain coordination in modular construction. Applied Sciences (Switzerland) 2021;11. https://doi.org/10.3390/app11135909.

[153] Li H, Yang D, Cao H, Ge W, Chen E, Wen X, et al. Data-driven hybrid petri-net based energy consumption behaviour modelling for digital twin of energy-efficient manufacturing system. Energy 2022;239. https://doi.org/10.1016/j.energy.2021.122178.

[154] Ma J, Chen H, Zhang Y, Guo H, Ren Y, Mo R, et al. A digital twin-driven production management system for production workshop n.d. https://doi.org/10.1007/s00170-020-05977-5/Published.

[155] Moi T, Cibicik A, Rølvåg T. Digital twin based condition monitoring of a knuckle boom crane: An experimental study. Eng Fail Anal 2020;112. https://doi.org/10.1016/j.engfailanal.2020.104517.

[156] Rolle RP, Martucci V de O, Godoy EP. Modular Framework for Digital Twins: Development and Performance Analysis. Journal of Control, Automation and Electrical Systems 2021;32:1485–97. https://doi.org/10.1007/s40313-021-00830-w.

[157] Zheng P, Sivabalan AS. A generic tri-model-based approach for product-level digital twin development in a smart manufacturing environment. Robot Comput Integr Manuf 2020;64. https://doi.org/10.1016/j.rcim.2020.101958.

[158] Wang K, Liu D, Liu Z, Wang Q, Tan J. An assembly precision analysis method based on a general part digital twin model. Robot Comput Integr Manuf 2021;68. https://doi.org/10.1016/j.rcim.2020.102089.

[159] Lim KYH, Zheng P, Chen CH, Huang L. A digital twin-enhanced system for engineering product family design and optimization. J Manuf Syst 2020;57:82–93. https://doi.org/10.1016/j.jmsy.2020.08.011.

[160] Leng J, Zhou M, Xiao Y, Zhang H, Liu Q, Shen W, et al. Digital twins-based remote semi-physical commissioning of flow-type smart manufacturing systems. J Clean Prod 2021;306. https://doi.org/10.1016/j.jclepro.2021.127278.

[161] Nikolakis N, Alexopoulos K, Xanthakis E, Chryssolouris G. The digital twin implementation for linking the virtual representation of human-based production tasks to their physical counterpart in the factory-floor. Int J Comput Integr Manuf 2019;32:1–12. https://doi.org/10.1080/0951192X.2018.1529430.

[162] Havard V, Jeanne B, Lacomblez M, Baudry D. Digital twin and virtual reality: a co-simulation environment for design and assessment of industrial workstations. Prod Manuf Res 2019;7:472–89. https://doi.org/10.1080/21693277.2019.1660283.

[163] Fan B, Wu Y, He Z, Chen Y, Quek TQS, Xu CZ. Digital Twin Empowered Mobile Edge Computing for Intelligent Vehicular Lane-Changing. IEEE Netw 2021;35:194–201. https://doi.org/10.1109/MNET.201.2000768.

[164] Maruyama T, Ueshiba T, Tada M, Toda H, Endo Y, Domae Y, et al. Digital twin-driven human robot collaboration using a digital human. Sensors 2021;21. https://doi.org/10.3390/s21248266.

[165] Corradini F, Silvestri M. Design and testing of a digital twin for monitoring and quality assessment of material extrusion process. Addit Manuf 2022;51. https://doi.org/10.1016/j.addma.2022.102633.

[166] Chaber P, Domański PD, Dąbrowski D, Ławryńczuk M, Nebeluk R, Plamowski S, et al. Digital Twins in the Practice of High-Energy Physics Experiments: A Gas System for the Multipurpose Detector. Sensors 2022;22. https://doi.org/10.3390/s22020678.

[167] Lu Y, Liu Z, Min Q. A digital twin-enabled value stream mapping approach for production process reengineering in SMEs. Int J Comput Integr Manuf 2021;34:764–82. https://doi.org/10.1080/0951192X.2021.1872099.

[168] Saroj AJ, Roy S, Guin A, Hunter M. Development of a Connected Corridor Real-Time Data-Driven Traffic Digital Twin Simulation Model. J Transp Eng A Syst 2021;147. https://doi.org/10.1061/jtepbs.0000599.

[169] Liu J, Liu J, Zhuang C, Liu Z, Miao T. Construction method of shop-floor digital twin based on MBSE. J Manuf Syst 2021;60:93–118. https://doi.org/10.1016/j.jmsy.2021.05.004.

[170] Garg G, Kuts V, Anbarjafari G. Digital twin for fanuc robots: Industrial robot programming and simulation using virtual reality. Sustainability (Switzerland) 2021;13. https://doi.org/10.3390/su131810336.

[171] Zhao H, Tan C, OBrien EJ, Zhang B, Uddin N, Guo H. Developing Digital Twins to Characterize Bridge Behavior Using Measurements Taken under Random Traffic. Journal of Bridge Engineering 2022;27. https://doi.org/10.1061/(asce)be.1943-5592.0001814.

[172] Mukhopadhyay A, Reddy GSR, Saluja KPS, Ghosh S, Peña-Rios A, Gopal G, et al. Virtual-reality-based digital twin of office spaces with social distance measurement feature. Virtual Reality and Intelligent Hardware 2022;4:55–75. https://doi.org/10.1016/j.vrih.2022.01.004.

[173] Guo H, Zhu Y, Zhang Y, Ren Y, Chen M, Zhang R. A digital twin-based layout optimization method for discrete manufacturing workshop. The International Journal of Advanced Manufacturing Technology 2021;112:1307–18. https://doi.org/10.1007/s00170-020-06568-0.

[174] Kuts V, Modoni GE, Otto T, Sacco M, Tähemaa T, Bondarenko Y, et al. Synchronizing physical factory and its digital twin throughan iiot middleware: A case study. Proceedings of the Estonian Academy of Sciences 2019;68:364–70. https://doi.org/10.3176/proc.2019.4.03.

[175] Moutis P, Alizadeh-Mousavi O. Digital Twin of Distribution Power Transformer for Real-Time Monitoring of Medium Voltage from Low Voltage Measurements. IEEE Transactions on Power Delivery 2021;36:1952–63. https://doi.org/10.1109/TPWRD.2020.3017355.

[176] Kang JS, Chung K, Hong EJ. Multimedia knowledge-based bridge health monitoring using digital twin. Multimed Tools Appl 2021;80:34609–24. https://doi.org/10.1007/s11042-021-10649-x.

[177] Bouzid S, Viarouge P, Cros J. Real-time digital twin of a wound rotor induction machine based on finite element method. Energies (Basel) 2020;13. https://doi.org/10.3390/en13205413.

[178] Redelinghuys AJH, Basson AH, Kruger K. A six-layer architecture for the digital twin: a manufacturing case study implementation. J Intell Manuf 2020;31:1383–402. https://doi.org/10.1007/s10845-019-01516-6.

[179] Polini W, Corrado A. Digital twin of composite assembly manufacturing process. Int J Prod Res 2020;58:5238–52. https://doi.org/10.1080/00207543.2020.1714091.

[180] Mo Y, Ma S, Gong H, Chen Z, Zhang J, Tao D. Terra: A Smart and Sensible Digital Twin Framework for Robust Robot Deployment in Challenging Environments. IEEE Internet Things J 2021;8:14039–50. https://doi.org/10.1109/JIOT.2021.3068736.

[181] Ward R, Soulatiantork P, Finneran S, Hughes R, Tiwari A. Real-time vision-based multiple object tracking of a production process: Industrial digital twin case study. Proc Inst Mech Eng B J Eng Manuf 2021;235:1861–72. https://doi.org/10.1177/09544054211002464.

[182] Bartos M, Kerkez B. Pipedream: An interactive digital twin model for natural and urban drainage systems. Environmental Modelling and Software 2021;144. https://doi.org/10.1016/j.envsoft.2021.105120.

[183] Mykoniatis K, Harris GA. A digital twin emulator of a modular production system using a data-driven hybrid modeling and simulation approach. J Intell Manuf 2021;32:1899–911. https://doi.org/10.1007/s10845-020-01724-5.

[184] Martínez-gutiérrez A, Díez-gonzález J, Ferrero-guillén R, Verde P, Álvarez R, Perez H. Digital twin for automatic transportation in industry 4.0. Sensors 2021;21. https://doi.org/10.3390/s21103344.

[185] Kalantari S, Pourjabar S, Xu TB, Kan J. Developing and user-testing a “Digital Twins” prototyping tool for architectural design. Autom Constr 2022;135. https://doi.org/10.1016/j.autcon.2022.104140.

[186] Wang SH, Tu CH, Juang JC. Automatic traffic modelling for creating digital twins to facilitate autonomous vehicle development. Conn Sci 2022;34:1018–37. https://doi.org/10.1080/09540091.2021.1997914.

[187] Fonseca ÍA, Gaspar HM, de Mello PC, Sasaki HAU. A Standards-Based Digital Twin of an Experiment with a Scale Model Ship. CAD Computer Aided Design 2022;145. https://doi.org/10.1016/j.cad.2021.103191.

[188] Cai Y, Wang Y, Burnett M. Using augmented reality to build digital twin for reconfigurable additive manufacturing system. J Manuf Syst 2020;56:598–604. https://doi.org/10.1016/j.jmsy.2020.04.005.

[189] Laaki H, Miche Y, Tammi K. Prototyping a Digital Twin for Real Time Remote Control over Mobile Networks: Application of Remote Surgery. IEEE Access 2019;7:20235–336. https://doi.org/10.1109/ACCESS.2019.2897018.

[190] Xiao J, Fan K. Research on the digital twin for thermal characteristics of motorized spindle. International Journal of Advanced Manufacturing Technology 2022;119:5107–18. https://doi.org/10.1007/s00170-021-08508-y.

[191] Ren J, Zhou T, Rong Y, Ma Y, Ahmad R. Feature-based modeling for industrial processes in the context of digital twins: A case study of HVOF process. Advanced Engineering Informatics 2022;51. https://doi.org/10.1016/j.aei.2021.101486.

[192] Li X, He B, Zhou Y, Li G. Multisource Model-Driven Digital Twin System of Robotic Assembly. IEEE Syst J 2021;15:114–23. https://doi.org/10.1109/JSYST.2019.2958874.

[193] Zhou C, Zhang F, Wei B, Lin Y, He K, Du R. Digital twin–based stamping system for incremental bending. The International Journal of Advanced Manufacturing Technology 2021;116:389–401. https://doi.org/10.1007/s00170-021-07422-7.

[194] Ward R, Sun C, Dominguez-Caballero J, Ojo S, Ayvar-Soberanis S, Curtis D, et al. Machining Digital Twin using real-time model-based simulations and lookahead function for closed loop machining control. The International Journal of Advanced Manufacturing Technology 2021;117:3615–29. https://doi.org/10.1007/s00170-021-07867-w.

[195] Schroeder GN, Steinmetz C, Rodrigues RN, Ventura Bayan Henriques R, Rettberg A, Pereira CE, et al. A Methodology for Digital Twin Modeling and Deployment for Industry 4.00, and proposes a methodology for DT design using model-driven engineering (MDE) that strives toward being both flexible and generic 2020. https://doi.org/10.13039/501100002322.

[196] Chaux JD, Sanchez-Londono D, Barbieri G. A digital twin architecture to optimize productivity within controlled environment agriculture. Applied Sciences (Switzerland) 2021;11. https://doi.org/10.3390/app11198875.

[197] Milton M, de La CO, Ginn HL, Benigni A. Controller-Embeddable Probabilistic Real-Time Digital Twins for Power Electronic Converter Diagnostics. IEEE Trans Power Electron 2020;35:9852–66. https://doi.org/10.1109/TPEL.2020.2971775.

[198] Steyn WJ, Broekman A. Development of a Digital Twin of a Local Road Network: A Case Study. J Test Eval 2022;50:20210043. https://doi.org/10.1520/jte20210043.

[199] Miao J, Lan S. Application of Visual Sensing Image Processing Technology under Digital Twins to the Intelligent Logistics System. Advances in Civil Engineering 2021;2021. https://doi.org/10.1155/2021/5743387.

[200] Saifutdinov F, Tolujevs J. Time and space discretization in the digital twin of the airport transport network. Transport and Telecommunication 2021;22:257–65. https://doi.org/10.2478/ttj-2021-0019.

[201] Yang W, Zheng Y, Li S. Digital twin of spacecraft assembly cell and case study. Int J Comput Integr Manuf 2022;35:263–81. https://doi.org/10.1080/0951192X.2021.1992657.

[202] Yablochnikov EI, Chukichev A v., Timofeeva OS, Abyshev OA, Abaev GE, Colombo AW. Development of an industrial cyber-physical platform for small series production using digital twins. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 2021;379. https://doi.org/10.1098/rsta.2020.0370.

[203] Liu C, le Roux L, Körner C, Tabaste O, Lacan F, Bigot S. Digital Twin-enabled Collaborative Data Management for Metal Additive Manufacturing Systems. J Manuf Syst 2022;62:857–74. https://doi.org/10.1016/j.jmsy.2020.05.010.

[204] Židek K, Pitel' J, Adámek M, Lazorík P, Hošovskỳ A. Digital twin of experimental smart manufacturing assembly system for industry 4.0 concept. Sustainability (Switzerland) 2020;12. https://doi.org/10.3390/su12093658.

[205] Chen G, Zhu J, Zhao Y, Hao Y, Yang C, Shi Q. Digital twin modeling for temperature field during friction stir welding. J Manuf Process 2021;64:898–906. https://doi.org/10.1016/j.jmapro.2021.01.042.

[206] Franciosa P, Sokolov M, Sinha S, Sun T, Ceglarek D. Deep learning enhanced digital twin for Closed-Loop In-Process quality improvement. CIRP Annals 2020;69:369–72. https://doi.org/10.1016/j.cirp.2020.04.110.

[207] Barni A, Pietraroia D, Züst S, West S, Stoll O. Digital twin based optimization of a manufacturing execution system to handle high degrees of customer specifications. Journal of Manufacturing and Materials Processing 2020;4. https://doi.org/10.3390/jmmp4040109.

[208] Aderiani AR, Wärmefjord K, Söderberg R, Lindkvist L. Individualizing locator adjustments of assembly fixtures using a digital twin. J Comput Inf Sci Eng 2019;19. https://doi.org/10.1115/1.4043529.

[209] Aivaliotis P, Georgoulias K, Chryssolouris G. The use of Digital Twin for predictive maintenance in manufacturing. Int J Comput Integr Manuf 2019;32:1067–80. https://doi.org/10.1080/0951192X.2019.1686173.

[210] Jain P, Poon J, Singh JP, Spanos C, Sanders SR, Panda SK. A digital twin approach for fault diagnosis in distributed photovoltaic systems. IEEE Trans Power Electron 2020;35:940–56. https://doi.org/10.1109/TPEL.2019.2911594.

[211] Seon G, Nikishkov Y, Makeev A, Ferguson L. Towards a digital twin for mitigating void formation during debulking of autoclave composite parts. Eng Fract Mech 2020;225. https://doi.org/10.1016/j.engfracmech.2019.106792.

[212] Ye Y, Yang Q, Yang F, Huo Y, Meng S. Digital twin for the structural health management of reusable spacecraft: A case study. Eng Fract Mech 2020;234. https://doi.org/10.1016/j.engfracmech.2020.107076.

[213] Yeratapally SR, Leser PE, Hochhalter JD, Leser WP, Ruggles TJ. A digital twin feasibility study (Part I): Non-deterministic predictions of fatigue life in aluminum alloy 7075-T651 using a microstructure-based multi-scale model. Eng Fract Mech 2020;228. https://doi.org/10.1016/j.engfracmech.2020.106888.

[214] Li Y, Wang S, Duan X, Liu S, Liu J, Hu S. Multi-objective energy management for Atkinson cycle engine and series hybrid electric vehicle based on evolutionary NSGA-II algorithm using digital twins. Energy Convers Manag 2021;230. https://doi.org/10.1016/j.enconman.2020.113788.

[215] Reed S, Löfstrand M, Andrews J. Modelling stochastic behaviour in simulation digital twins through neural nets. Journal of Simulation 2022;16:512–25. https://doi.org/10.1080/17477778.2021.1874844.

[216] Zhang T, Liu X, Luo Z, Dong F, Jiang Y. Time series behavior modeling with digital twin for Internet of Vehicles. EURASIP J Wirel Commun Netw 2019;2019. https://doi.org/10.1186/s13638-019-1589-8.

[217] Alexopoulos K, Nikolakis N, Chryssolouris G. Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing. Int J Comput Integr Manuf 2020;33:429–39. https://doi.org/10.1080/0951192X.2020.1747642.

[218] Kender R, Kaufmann F, Rößler F, Wunderlich B, Golubev D, Thomas I, et al. Development of a digital twin for a flexible air separation unit using a pressure-driven simulation approach. Comput Chem Eng 2021;151. https://doi.org/10.1016/j.compchemeng.2021.107349.

[219] Pilati F, Tronconi R, Nollo G, Heragu SS, Zerzer F. Digital twin of covid-19 mass vaccination centers. Sustainability (Switzerland) 2021;13. https://doi.org/10.3390/su13137396.

[220] Yang J, Son YH, Lee D, do Noh S. Digital Twin-Based Integrated Assessment of Flexible and Reconfigurable Automotive Part Production Lines. Machines 2022;10. https://doi.org/10.3390/machines10020075.

[221] Zou R, Liang X, Chen Q, Wang M, Zaghloul MAS, Lan H, et al. A Digital Twin Approach to Study Additive Manufacturing Processing Using Embedded Optical Fiber Sensors and Numerical Modeling. Journal of Lightwave Technology 2020;38:6402–11. https://doi.org/10.1109/JLT.2020.3010722.

[222] Liu Z, Chen W, Zhang C, Yang C, Chu H. Data Super-Network Fault Prediction Model and Maintenance Strategy for Mechanical Product Based on Digital Twin. IEEE Access 2019;7:177284–96. https://doi.org/10.1109/ACCESS.2019.2957202.

[223] Defraeye T, Tagliavini G, Wu W, Prawiranto K, Schudel S, Assefa Kerisima M, et al. Digital twins probe into food cooling and biochemical quality changes for reducing losses in refrigerated supply chains. Resour Conserv Recycl 2019;149:778–94. https://doi.org/10.1016/j.resconrec.2019.06.002.

[224] Delussu F, Manzione D, Meo R, Ottino G, Asare M. Experiments and Comparison of Digital Twinning of Photovoltaic Panels by Machine Learning Models and a Cyber-Physical Model in Modelica. IEEE Trans Industr Inform 2022;18:4018–28. https://doi.org/10.1109/TII.2021.3108688.

[225] Stavropoulos P, Papacharalampopoulos A, Athanasopoulou L. A molecular dynamics based digital twin for ultrafast laser material removal processes. International Journal of Advanced Manufacturing Technology 2020;108:413–26. https://doi.org/10.1007/s00170-020-05387-7.

[226] dos Santos CH, Gabriel GT, do Amaral JVS, Montevechi JAB, de Queiroz JA. Decision-making in a fast fashion company in the Industry 4.0 era: a Digital Twin proposal to support operational planning. The International Journal of Advanced Manufacturing Technology 2021;116:1653–66. https://doi.org/10.1007/s00170-021-07543-z.

[227] He R, Chen G, Dong C, Sun S, Shen X. Data-driven digital twin technology for optimized control in process systems. ISA Trans 2019;95:221–34. https://doi.org/10.1016/j.isatra.2019.05.011.

[228] Sweat R, Park JG, Liang R. A digital twin approach to a quantitative microstructure-property study of carbon fibers through HRTEM characterization and multiscale FEA. Materials 2020;13. https://doi.org/10.3390/MA13194231.

[229] Chetan M, Yao S, Griffith DT. Multi-fidelity digital twin structural model for a sub-scale downwind wind turbine rotor blade. Wind Energy 2021;24:1368–87. https://doi.org/10.1002/we.2636.

[230] Wang Y, Tao F, Zhang M, Wang L, Zuo Y. Digital twin enhanced fault prediction for the autoclave with insufficient data. J Manuf Syst 2021;60:350–9. https://doi.org/10.1016/j.jmsy.2021.05.015.

[231] Coito T, Martins MSE, Firme B, Figueiredo J, Vieira SM, Sousa JMC. Assessing the impact of automation in pharmaceutical quality control labs using a digital twin. J Manuf Syst 2022;62:270–85. https://doi.org/10.1016/j.jmsy.2021.11.014.

[232] Gejo-García J, Reschke J, Gallego-García S, García-García M. Development of a System Dynamics Simulation for Assessing Manufacturing Systems Based on the Digital Twin Concept. Applied Sciences (Switzerland) 2022;12. https://doi.org/10.3390/app12042095.

[233] Aubert K, Germaneau A, Rochette M, Ye W, Severyns M, Billot M, et al. Development of Digital Twins to Optimize Trauma Surgery and Postoperative Management. A Case Study Focusing on Tibial Plateau Fracture. Front Bioeng Biotechnol 2021;9. https://doi.org/10.3389/fbioe.2021.722275.

[234] Zhou Y, Xing T, Song Y, Li Y, Zhu X, Li G, et al. Digital-twin-driven geometric optimization of centrifugal impeller with free-form blades for five-axis flank milling. J Manuf Syst 2021;58:22–35. https://doi.org/10.1016/j.jmsy.2020.06.019.

[235] Guan J, Li Y, Liu J, Duan X, Shen D, Jia D, et al. Experimental and numerical research on the performance characteristics of OPLVCR engine based on the NSGA II algorithm using digital twins. Energy Convers Manag 2021;236. https://doi.org/10.1016/j.enconman.2021.114052.

[236] Xia M, Shao H, Williams D, Lu S, Shu L, de Silva CW. Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning. Reliab Eng Syst Saf 2021;215. https://doi.org/10.1016/j.ress.2021.107938.

[237] Wilson R, Mercier P, Patarachao B, Navarra A. Partial least squares regression of oil sands processing variables within discrete event simulation digital twin. Minerals 2021;11. https://doi.org/10.3390/min11070689.

[238] Ademujimi T, Prabhu V. Digital Twin for Training Bayesian Networks for Fault Diagnostics of Manufacturing Systems. Sensors 2022;22. https://doi.org/10.3390/s22041430.

[239] Savolainen J, Urbani M. Maintenance optimization for a multi-unit system with digital twin simulation: Example from the mining industry. J Intell Manuf 2021;32:1953–73. https://doi.org/10.1007/s10845-021-01740-z.

[240] Gonzalez M, Salgado O, Croes J, Pluymers B, Desmet W. A Digital Twin for Operational Evaluation of Vertical Transportation Systems. IEEE Access 2020;8:114389–400. https://doi.org/10.1109/ACCESS.2020.3001686.

[241] Yu J, Liu P, Li Z. Hybrid modelling and digital twin development of a steam turbine control stage for online performance monitoring. Renewable and Sustainable Energy Reviews 2020;133. https://doi.org/10.1016/j.rser.2020.110077.

[242] Sieber I, Thelen R, Gengenbach U. Enhancement of high-resolution 3d inkjet-printing of optical freeform surfaces using digital twins. Micromachines (Basel) 2021;12:1–12. https://doi.org/10.3390/mi12010035.

[243] Kuritsyn DN, Kuritsyna V v., Siluyanova M v. Digital Twins in the Design of Tools for Friction Stir Welding. Russian Engineering Research 2021;41:357–9. https://doi.org/10.3103/S1068798X21040146.

[244] Ghosh AK, Ullah AS, Kubo A. Hidden Markov model-based digital twin construction for futuristic manufacturing systems. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 2019;33:317–31. https://doi.org/10.1017/S089006041900012X.

[245] Guivarch D, Mermoz E, Marino Y, Sartor M. Creation of helicopter dynamic systems digital twin using multibody simulations. CIRP Annals 2019;68:133–6. https://doi.org/10.1016/j.cirp.2019.04.041.

[246] Grieves M. Product lifecycle management: The new paradigm for enterprises. International Journal Product Development 2005:71–84.

[247] Tao F, Zhang M, Liu Y, Nee AYC. Digital twin driven prognostics and health management for complex equipment. Cirp Annals-Manufacturing Technology 2018;67:169–72.

[248] Glaessgen E, Stargel D. The digital twin paradigm for future NASA and US Air Force vehicles. 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA, 2012, p. 1818.

[249] Friederich J, Francis DP, Lazarova-Molnar S, Mohamed N. A framework for data-driven digital twins of smart manufacturing systems. Comput Ind 2022;136:103586. https://doi.org/https://doi.org/10.1016/j.compind.2021.103586.

[250] Shao G, Jain S, Laroque C, Lee LH, Lendermann P, Rose O. Digital twin for smart manufacturing: the simulation aspect. 2019 Winter Simulation Conference (WSC), IEEE; 2019, p. 2085–98.

[251] Park KT, Lee J, Kim HJ, Noh S Do. Digital twin-based cyber physical production system architectural framework for personalized production. International Journal of Advanced Manufacturing Technology 2020;106:1787–810. https://doi.org/10.1007/s00170-019-04653-7.

[252] Palisade. Guide to Using @RISK Risk Analysis and Simulation Add-In for Microsoft ® Excel. 2004.

[253] Segovia M, Garcia-Alfaro J. Design, Modeling and Implementation of Digital Twins. Sensors 2022;22. https://doi.org/10.3390/s22145396.

[254] Steindl G, Stagl M, Kasper L, Kastner W, Hofmann R. Generic Digital Twin Architecture for Industrial Energy Systems. Applied Sciences 2020;10. https://doi.org/10.3390/app10248903.

[255] Alves RG, Souza G, Maia RF, Tran ALH, Kamienski C, Soininen J-P, et al. A digital twin for smart farming. 2019 IEEE Global Humanitarian Technology Conference (GHTC), IEEE; 2019, p. 1–4.

[256] Uhlenkamp J-F, Hribernik K, Wellsandt S, Thoben K-D. Digital Twin Applications: A first systemization of their dimensions. 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), IEEE; 2019, p. 1–8.

[257] Duda O, Kunanets N, Matsiuk O, Pasichnyk V. Cloud-based IT Infrastructure for “Smart City” Projects. Dependable IoT for Human and Industry, River Publishers; 2022, p. 389–409.

[258] Alphonsus ER, Abdullah MO. A review on the applications of programmable logic controllers (PLCs). Renewable and Sustainable Energy Reviews 2016;60:1185–205. https://doi.org/10.1016/j.rser.2016.01.025.

[259] Agust, Rullhn. Programmable Logic Controllers versus Personal Computers for Process Control. vol. 33. 1997.

[260] Dunning GA. Introduction to programmable logic controllers. Cengage Learning; 2005.

[261] Parr EA. Programmable controllers: an engineer’s guide. Newnes; 2003.

[262] Boje C, Guerriero A, Kubicki S, Rezgui Y. Towards a semantic Construction Digital Twin: Directions for future research. Autom Constr 2020;114:103179. https://doi.org/https://doi.org/10.1016/j.autcon.2020.103179.